The Pennsylvania State University
The Graduate School
College of Information Sciences and Technology

COMPUTATIONAL MODELING OF COMPOSITIONAL AND RELATIONAL DATA USING OPTIMAL TRANSPORT AND PROBABILISTIC MODELS

A Dissertation in
Information Sciences and Technology
by
Jianbo Ye

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2018

The dissertation of Jianbo Ye was reviewed and approved* by the following:

James Z. Wang
Professor of Information Sciences and Technology
Dissertation Co-Adviser, Co-chair of Committee

Jia Li
Professor of Statistics
Dissertation Co-Adviser, Co-chair of Committee

C. Lee Giles
Professor of Information Sciences and Technology

Zhenhui Jessie Li
Associate Professor of Information Sciences and Technology

Reginald B. Adams
Associate Professor of Psychology

Andrea Tapia
Associate Professor of Information Sciences and Technology
Graduate Program Chair

*Signatures are on file in the Graduate School.

Abstract

Quantitative researchers often view our world as a large collection of data generated and organized by the structures and functions of society and technology. Those data are usually presented and accessed with hierarchies, compositions, and relations. Understanding the structures and functions behind such data requires using models and methods for specifically analyzing their associated data structures. One of the biggest challenges in achieving this goal is developing a *principled* data and model framework capable of meaningfully exploiting the structured knowledge of data. Those structures of data include compositional and relational patterns: multiple entities have to interact and group in order to make sense. Although the conventional vector-based data analysis pipelines have become the standard quantitative framework for many fields in sciences and technology, they are not directly applicable to and have several limitations for extracting knowledge from compositional and relational data.

The goal of this thesis research is to introduce new mathematical models and computational methods for analyzing large-scale compositional and relational data, as well as to validate the models' usefulness in solving real-world problems. We begin by introducing several backgrounds, including optimal transport, an old but refreshing topic in mathematics, and probabilistic graphical model, a

popular tool in statistical modeling. Particularly, we explain how optimal transport relates to an important modeling concept, *a.k.a. matching*, in machine learning. Next, we present our work related to computational algorithms of those relational and structural models including a fast discrete distribution clustering method using Wasserstein barycenters, a simulated annealing-based inexact oracle for Wasserstein loss minimization, a Bregman ADMM-based oracle for Wasserstein geodesic classification, and a probabilistic multi-graph model for consensus analysis. Their computational complexities, numerical difficulties, scalability, and accuracy issues are discussed in depth. We apply those computational algorithms to several areas, such as document analysis and crowdsourcing, by treating data as relational quantities from a perspective that has not been fully studied in the literature. We will conclude by discussing challenges in developing suitable methods for compositional and relational data and review more recent work that addresses several past concerns.

Table of Contents

List of Figures

List of Tables

Acknowledgments

Many people have walked alongside me during the past five years. They have been with me, inspired me, and helped shape my life in these years. I offer my sincere gratitude and appreciation to each of them.

I would especially like to thank my advisors Dr. James Z. Wang and Dr. Jia Li. Working with these two scholars and researchers has been a fantastic experience. They have guided me to become a mature researcher who is competent in developing meaningful and innovative research with independent thinking. Under their guidance, I have had the opportunity and freedom to study machine learning to a great extent, to investigate many interesting and deep sub-problems in AI, and to define my own research. With their encouragement, I have developed the courage and optimism to face the most difficult situations of a junior researcher. This precious experience not only materializes my professionalism but also shapes my attitudes and disciplines.

I also thank Dr. Zhenhui (Jessie) Li, Dr. Reginald B. Adams, and Dr. C. Lee Giles for serving on my committee as for providing additional insights and support for my thesis. Dr. Reginald B. Adams and Dr. Michelle G. Newman gave insightful discussion and invaluable guidance from a psychological perspective, and I am deeply grateful for them. Dr. Jose A. Piedra-Fernández generously helped when

I visited Universidad de Almería, Spain in the summer of 2016, and I offer my sincere thanks. I also thank Dr. Xin Lu and Dr. Zhe Lin for their great support and mentorship when I worked at Adobe System Inc. in the summer of 2017.

My gratitude also extends to my peers and colleagues within and outside Penn State. I am indebted to Xin Lu, Yu Zhang, Baris Kandemir, Hanjoo Kim, Chen Liang, Yukun Chen, Xinye Zheng, Luo Yu, Zhuomin Zhang, Panruo Wu, Min Yang, Yanran Li, and Stephen Wistar for our joint efforts on research projects and papers. I am also grateful to Jian Wu, Zhaohui Wu, Mingyi Zhao, Fei Wu, Hongjian Wang, Pinyao Guo, Xiao Liu, Alexander G. Ororbia II, Mohammad Kamani, and Farshid Farhat for inspiring discussions.

My Ph.D. study was funded by the National Science Foundation and the College of Information Sciences and Technology. The findings and conclusions do not necessarily reflect the view of the funding agencies. I am truly grateful for the support. Finally, I would like to thank my family and friends for their continued support.

This dissertation is dedicated to my family.

Chapter 1
Introduction

Our world produces data in a variety of forms in terms of structure and function. Those data are usually presented as hierarchies, compositions, and relations. The growing diversity of these forms raises the vast need to create suitable and advanced analytic methods through modeling and computation. Bladesmithing is the art of making blades such as knives, swords, and daggers using smithing tools provides an interesting analogy for understanding the need for various methods for analyzing data. Blades are historically made for different purposes, namely to cut objects for various purposes. Those uses require different degrees of thickness, shape, and tensile strength. Similarly in the era of big data, we need models for understanding different types of data, with a diverse set of purposes subject to the facts, *e.g.* where data come from, how data are generated, why data are collected, etc.

As a modeler, I understand the temptation to treat data in a consistent manner. To convert data into familiar representations and then apply our existing modeling skills to extract information from them is somewhat comforting. For example, one who works in topic modeling often converts data into tokenized counts before proceeding to extract modes of data as topics. One who works in convolutional neural networks often converts data into vectors that inherit some spatial-temporal patterns before proceeding to build a predictive model. Thinking as a bladesmith, one has to know what types of blades is most suitable to do the job by re-positioning the purpose in a priority. Such decision-making requires critical understanding of the properties of the target. In this thesis, we present our approach and work as a bladesmith's striving to hone a more suitable blade: specifically, we build computational models tailored to subjects.

In other words, this thesis is devoted to modeling compositional and relational

data with unconventional approaches. Despite being unconventional, these approaches are arguably often more natural in the sense of collecting and generating data. Modelers are always subject to certain assumptions about data, but some assumptions exist for computational convenience rather than the simplification of data composition or relations. With the advance of modern computer architectures, however, we become more flexible in terms of computations. We no longer need to follow some conventions of past practices that developed from constraints of hardware or software limitations.

Developing such unconventional approaches as shown in this thesis requires innovation from the most basic grounds for data and pattern representation, mastery of some foundational mathematics and solid computational methods to support the rigor and the feasibility, and real world opportunities where those approaches become valuable. This thesis focuses on those tasks. As an overview, we first provide a holistic picture of my thesis research in Chapter 1 including several basic concepts, main ideas and challenges, and potential impacts. Next, we introduce a "sword" — optimal transport (OT) — as a key concept to tackle the research challenge. Particularly, we present some prerequisites for rigorously understanding optimal transport and the motivations for developing OT models in Chapter 2. Most notations are established for reference in later chapters. Next, we present a cookbook for how the "sword" is made by listing three scalable approaches that computationally solve optimal transport. This thesis research contributes to the development of two approaches. Therefore, we provide more details about them, including algorithms, implementations, applications, and experiments in relevant chapters. Almost all OT models use one of those approaches as a backbone. A comprehensive review of other techniques is offered in Chapter 3. We demonstrate their use in Chapter 4, where models built on top of optimal transport become computationally tractable using the numerical methods in Chapter 3. The experiment results show the advantages of using Wasserstein geometry to model the data over vector-based models, and how specific approximation strategy leads to different effects. In Chapter 5, we present a specific application of D2-Clustering in text analysis domain, showing the potential of this new family of models in practice. After finalizing the chapters related to OT, we then describe a graph-based approach for crowdsourced affective data in Chapter 6. Graph, as a natural way to represent relational quantities, is a popular way of modeling. The method developed

in the thesis overcomes the limitation of conventional vector-based approaches in improving the quality of crowdsourced affective data.

1.1 Several Key Concepts in Modeling

This section offers a generalized review of several key concepts in modeling. Those who are familiar with the literature may skip this section. We found this part to be essential to most audiences who are new to the subfield, even for experienced machine-learning researchers and practitioners. Since we mainly discuss non-vector approaches in the thesis, it is important to track the model assumptions, identify parameters (representation in the continuous and discrete sense), and understand how they are computed or updated numerically.

Geometric models. A geometric model is an abstraction of data in metric space. The geometry of data basically gives a distance measurement between any two data "points" in the sample space. Here "point" refers to any type of entity in the dataset. It could be a single point in a Euclidean space or a single distribution. The goal of a geometry model is to extract part of the geometric information encoded in the data.

Probabilistic models. A probabilistic model is an abstraction of data that are hypothetically treated as samples from a probabilistic model. A probabilistic model is a generative model if the data are hypothetically generated in a completely specified way but is a discriminative model if only part of the data are hypothetically generated by observing the other part of data.

Vector-based models. A vector based model represents the entity of data as a vector, whose quantities are indexed by the dimensions. For example, an image can be represented by a vector of pixels, whose length is width times height, by ignoring the locational annotations of pixels.

Geometric distributional models. A geometric distributional model is a geometric model for distributions. Precisely speaking, a set of distributions is embodied with a metric space or a measurement structure such that a geometric model can be constructed. In fact, the thesis will put a large body on Wasserstein space, a space derived from the optimal transport principle. Wasserstein space is a metric space for distributions with a metricized support set.

Graph models. A graph model is an abstraction of graph data. It could be

a probabilistic generative model by specifying how the graphs (a.k.a. nodes and edges) are created. A multi-graph is a set of graphs which have shared nodes.

Parameters. Parameters are numerical quantities that control the variations of a model family. In fact, it does not have to be vectors or scalars. In some of the distributional models explored in the thesis, the parameters are realized as the discretization of a distribution or the discrete distribution.

Algorithms. When a model is fitted to data, its parameters are estimated numerically by an algorithm. Such algorithms are computational methods that instantiate a model representing a particular dataset, and can be implemented using modern computers.

1.2 Innovation, Gap, and Impacts

The innovation of this research starts from the basic representation of entities. Classical methodologies in machine learning and data analytics often assume a vector-based representation of entities. That is, a fixed-length vector is used to summarize useful aspects of each entity. Vector-based representation is often convenient for computation and storage purposes, but it is not always justified for understanding real-world sophisticated situations, where unstructured data are ubiquitous. When each entity is a complex object containing information at multiple levels (*e.g.*, images, documents, sequences, and graphs), it becomes significantly nontrivial to construct such vector representation. In many quantitative disciplines, data and domain experts must manually construct such a vector representation. Since such process is open-ended and requires domain knowledge, useful information can be lost in the preprocessing stage when original data is forced into a certain format.

This research proposes to represent such entities by sets of unordered and weighted components and we call this data model *compositional*. We develop a school of machine learning methods for this data model without converting them into vectors. The methods include clustering, gradient-based learning, component analysis, and classification. By learning directly from compositional data, we bypass the need to extract high-level features, overcoming the limitations of past understandings of data when only a finite number of features are studied. Moreover, patterns discovered from compositional data are itself compositional, hence are

highly interpretable and easy to diagnose. The core technologies developed in this research are used to analyze massive, distributed data in numerous areas of sciences and the industry.

Another line of this thesis research is to propose a new probabilistic modeling approach for analyzing crowdsourced affective data [7]. Crowdsourcing is an emerging direction for collecting large-scale psychological data using the Internet. The goal is to identify more reliable or informative data annotators in the subject pool of crowdsourcing and eventually to improve the quality of collected data. One innovation of this research is the conversion of data (labels provided by workers) to relational quantities (essentially building an agreement multi-graph) before modeling them. The relational representation of crowdsourced data also let one avoid making several unrealistic assumptions as are done in conventional models.

Revolutionizing the way data are represented is only the first step. The second area of this research adopts the concepts and geometric understandings from the optimal transport theory to develop mathematically rigorous models, and pushes these models to tractable computational solutions. Concretely, each compositional entity is treated mathematically as an empirical measure on a metric space. The Wasserstein geometry is then used to characterize the space structures and the proximity of measures. Established machine learning models were originally developed for data sitting in the Euclidean space, but rarely for the Wasserstein space. The thesis comprehensively reviews the recent computational efforts of the machine learning community in generalizing conventional models for the Wasserstein space, and as a novel work in this field, this research proposes a set of new models and methods following this spirit. In order to develop tractable algorithms for this new family of models, a diverse set of computational methodologies are explored and innovated in this research, crossing several subfields in numerical optimization and computational statistics. The major challenge of realizing this goal originates from the fact that the Wasserstein distance does not have a closed form to calculate, but is instead defined by a variational formulation. Computing Wasserstein distance can be done by linear programming at the complexity of $O(n^3 \log n)$, whereas computing other common discrepancy measures in machine learning is extremely cheap, only at the complexity of $O(n)$. The high level of complexity has prevented machine learning researchers from studying the usefulness of Wasserstein distance

in creating practical models for years. This thesis research develops two families of approaches that one can actually approximate Wasserstein distance using $O(n^2/\varepsilon)$ algorithms, substantially overcoming the aforementioned difficulty in developing machine learning models for compositional data. The technical work of this research have been published in premier journals and conferences in the field [8–11].

In the third area of my thesis research, we implement a set of computational toolboxes that leverage current state-of-the-art parallel computing infrastructures to realize data analytics at scale. It has been empirically successful in analyzing a variety of real world data. For example, it was reported that the D2-Clustering toolbox can parallel process millions of discrete distributions using hundreds of CPU cores with over 80% scaling efficiency. The related technologies have been filed as US patent and have already been used in several research projects in computer vision [12], Bayesian statistics [13], computational linguistics [9], and meteorological imaging science [14]. Another example is the tool developed for improving the quality of crowdsourced affective data. A US patent is being filed for this technology. Another doctoral student of the research group is using the technology to systematically measure and control data quality for crowdsourcing emotion-related data.

Chapter 2
Optimal Transport: Background

2.1 Mathematical Backgrounds

We start from the basic concept of optimal transport. Based on their computational natures, one can categorize the OT problems into continuous ones and discrete ones. Here are two examples.

Continuous Optimal Transport. Please see Fig. 2.1. Assume that we are given a pile of sand, and a hole that we have to completely fill up with the sand. Certainly, we have to make sure the pile and the hole must have the same volume. Without loss of generality, we assume the mass of the pile is 1. We shall model both the pile and the hole by probability measures μ, ν, defined respectively on some measure spaces X, Y. Whenever A and B are measurable subsets of X and Y respectively, $\mu(A)$ gives the amount of sand located inside A; and $\nu(B)$ gives the amount of sand to be piled inside B.

Moving sand from space X to space Y takes some effort, which is modeled by a measurable cost function defined on $X \times Y$. Informally, $c(x, y) : X \times Y \mapsto \mathbb{R}_+ \cup +\infty$ tells how much it costs to transport one unit of mass from location $x \in X$ to location $y \in Y$. It is natural to assume c is measurable and nonnegative, and can take the infinity.

A basic question is how to transport sand from μ to ν at minimal cost.

Discrete Optimal Transport. Assume that we are transporting wood from several wood processing plants to several factories producing wood products. Each plant has a certain quantity of wood already collected from mining sites, and each

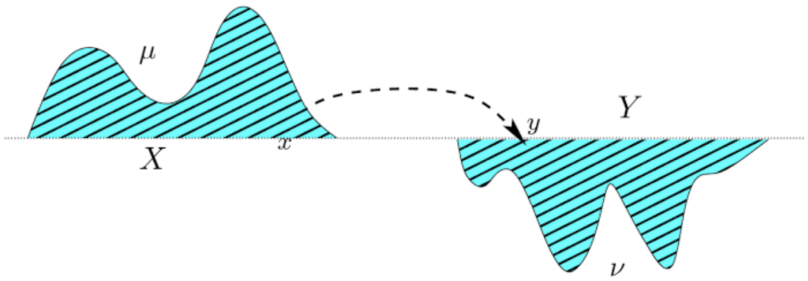


Figure 2.1. Moving sands, the figure is from Villani's book [1]

factory has certain needs. One can represent the quantities at plants as vector $\mathbf{p} \in \mathbb{R}_+^{m_1}$, and the quantities needed at factories as vector $\mathbf{q} \in \mathbb{R}_+^{m_2}$. Similar to our sand problem, the total quantities of produced wood at plants and needed wood at factories are the same, that is $\langle \mathbf{p}, \mathbb{1} \rangle = \langle \mathbf{q}, \mathbb{1} \rangle = 1$.

Transportation of wood also needs some effort, which is modeled by a cost matrix $M \in \mathbb{R}_+^{m_1 \times m_2}$. Each element $M_{i,j}$ in the cost matrix represents the transportation cost between pair of plant i and factory j indexed by its row and column respectively.

Another basic question is how to develop a transportation plan between plants and factories, by specifying in a pairwise manner how much wood from one plant should be carried to one factory, such that the total cost is minimal. After such a minimum cost plan is found, one may visualize the plan which would look like something displayed in Fig. 2.2.

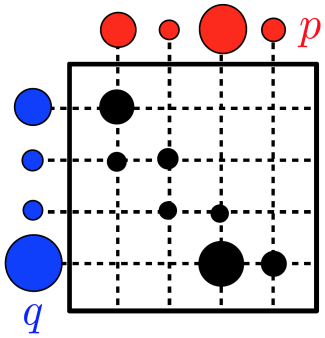


Figure 2.2. The sizes of circles visualizes the quantities of wood. Summations of quantities of all black circles by row should be equal to quantities of the blue circles, while summations by column should be equal to that of red.

There are many variants of OT problems depending on how the "distribution" is defined. In the sand example, a continuous distribution (or a probability measure)

is used to model a pile of sand. In the wood example, a discrete distribution is used instead to model how wood can be distributed across different plants or factories. A more common setup is to consider modeling data as "bags of vectors", where each vector in bag is sitting at the same Euclidean space. By setting the so-called cost between any two vectors to their Euclidean distance, one can define a metric distance between any two "bags" as their minimal transportation cost using a similar OT formulation.

The OT formulation not only gives a way to measure the distance/dissimilarity between compositional data, a.k.a. bags, but also gives an interpretable transportation plan. We will call such plan as "matching" in the sequel. We want to emphasize that "matching" is a powerful idea as a modeling tool, because it is decided dynamically between two collections of sand/woord. Under proper assumptions, matching is deterministic as it follows the principle of OT. We will revisit this concept many times in later chapters.

In probability theory, Wasserstein distance is a geometric distance naturally defined for any two probability measures over a metric space.[1]

Definition 2.1.1 (p-Wasserstein distance)**.** Given two probability distribution μ, ν defined on Euclidean space $\mathbb{R}^d$, the p-Wasserstein distance $W_p(\cdot, \cdot)$ between them is given by

$$W_p(\mu, \nu) \stackrel{\text{def.}}{=} \left[\inf_{\gamma \in \Pi(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|\mathbf{x} - \mathbf{y}\|^p d\gamma(\mathbf{x}, \mathbf{y}) \right]^{1/p}, \tag{2.1}$$

where $\Pi(\mu, \nu)$ is the collection of all distributions on $\mathbb{R}^d \times \mathbb{R}^d$ with marginal f and g on the first and second factors respectively. In particular, the $\Pi(\cdot, \cdot)$ is often called the coupling set. The $\gamma^* \in \Pi(\mu, \nu)$ that takes the infimum in Eq. (2.1) is called the optimal coupling.

Remark 1. By the Hölder inequality, one has $W_p \leq W_q$ for any $p \leq q < \infty$. In this paper, we focus on the practice of W_p with $0 < p \leq 2$.

[1] The name "Wasserstein distance" was coined by R. L. Dobrushin in 1970, after the Russian mathematician Leonid Vaserštein who introduced the concept in 1969. Most English-language publications use the German spelling "Wasserstein" (attributed to the name "Vaserstein" being of German origin).

2.2 Motivations for Developing OT Models

Before we study the mathematics of optimal transport, we provide some application background for using optimal transport or matching in pattern modeling.

2.2.1 Computer Vision and Imaging Science

With the growing volume of image and video data in the world today and tremendous diversity of applications in which images are involved, the need for generic methods to deal with this huge amount of data is rapidly increasing. A common school of practices begins by gathering statistics from an image as a feature representation, and then using feature analytical methods to extract meaningful patterns from those statistics. Because an image is considered a complex object, there are many ways to extract such statistics. Thus, relevant features such as color, contours, orientations or textures are first extracted from images and are then computationally represented in *histograms* — a popularly used representation in image processing. Histograms are built from the quantization of the space of features into discrete bins. This process recasts the problems associated with image diversity and allows the retrieval of a "similar" image in a database or segmentation of a particular object in an image by comparing histograms. There are many efforts in the computer vision and imaging community to develop advanced ways to enhance image descriptors (including global descriptors, keypoint descriptors, and patch descriptors) and to improve varying aspects of their robustness. Yet the common metrics used for their comparison are still not sufficient to deal with various perturbation effects (such as shifting, quantization, and deformation). In fact, this topic has long been pursued in signal processing, where numerous researchers are attempting to address the issue using data-driven approaches.

The optimal transport framework provides a compact and unified way for addressing the robustness of histogram comparison but requires significant computational cost. It was first considered for image comparison in 1989 [15]. Later, it was introduced to computer science community as the Earth Mover's Distance (EMD) [16], where an image retrieval system was developed based on the similarity between color histograms extracted from image data. Since then, EMD has been established as an indispensable role in computer vision and multimedia retrieval,

	$x_1 =$ 	$x_2 =$ 
$\|x - x_0\|_2$	5.89	5.07
$\mathrm{KL}\left(x \middle\| \frac{x+x_0}{2}\right)$	0.36	0.28
$W_2(x, x_0)$	3.38	4.45

Table 2.1. Different metrics in comparing two images

where a distribution-valued signature (including histogram) is adopted for object representation [17–19].

The following example may help clarify the motivation for using optimal transport for histogram comparison:

Example 2.2.1. Given $x_0 =$  , which one of the two is more similar (or closer) to the query image? $x_1 =$  or $x_2 =$  ?

We compute different metrics for comparing two images in Table 2.1. The first method treats the image as a vector of pixel intensities and directly calculates their Euclidean distance. The second method treats the image as a 2D (normalized) histogram and calculates the KL divergence. The last method again treats the image as a 2D histogram but calculates the 2-Wasserstein distance (See Definition 2.1).

To understand the reason why the regular metrics (Euclidean distance or KL divergence) implies the second image (the digit "3") is closer or more similar to the query image than the first image (the digit "2"), we must figure out what is missing in the computation of those metrics.

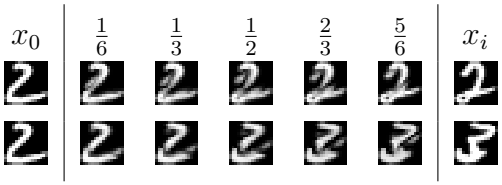
By shuffling the locations of pixels and redisplay their intensities via an image (See Matlab code below), can we correctly compare two images?

```
x0=vec( ); x1=vec( ); x2=vec( );
idx=randperm(length(x0));
x0=x0(idx); x1=x1(idx); x2=x2(idx);
```

The original images and redisplayed images are shown as follows:

It is impossible for a human to compare redisplayed images at the second row. It is clear that not only the intensity of pixels but also the location of pixels

contribute to our visual perception of image similarity. However, the regular metrics (such as Euclidean distance, KL divergence) stay unchanged to any pair of the redisplayed images, because they do not use the pixels' locations in their calculations. 2-Wasserstein distance is however a metric sensitive to the geometry of pixels by its definition. It calculates the amount of effort in (minimum) transporting pixel intensities between images on the 2D plane. If we visualize the intermediate stages of this optimal transportation subject to an interpolating factor $t = \frac{i}{6}$ with $i = 1, \ldots, 5$, we may see images like below:



To this extent, most will see that the visualization of how the optimal transportation between pair (x_0, x_1) or (x_0, x_2) is conducted matches our intuition of comparing two histograms.

One may argue that there are many data-driven approaches that seek a proper similarity measure for comparing complex objects. Data-driven approaches are mostly successful when a test pair has sufficient evidences or clues in the training set that tell how similar or dissimilar they are. In other words, a proper similarity measure can be learned from previous experiences. If we believe real-world data are sampled from a manifold (an assumption may be flawed in its own), the dissimilarity between two points should be their geodesic distance on the manifold (See Fig. 2.3). The comparison made based on the principle of optimal transport on the other hand is following a different perspective. Instead of relying on previous experience and a learning paradigm, the OT framework relies on the source of the prior knowledge about data in order to create a matching problem. In our case, the prior knowledge is the location of pixels. The prior knowledge often tells how to meaningfully interpret data that is not directly expressed in data itself. It therefore becomes an edge, which other conventional approaches never have the opportunity to consider. We will revisit this same philosophy in the next section.

Another interesting fact of using OT as a basic principle in building a space of histograms is the Wasserstein barycenter problem, discussed further in Chapter 4. In the Wasserstein space, we actually can define a type of "centroid" in an analog to the mean in Euclidean space. This special "centroid" possesses an intriguing

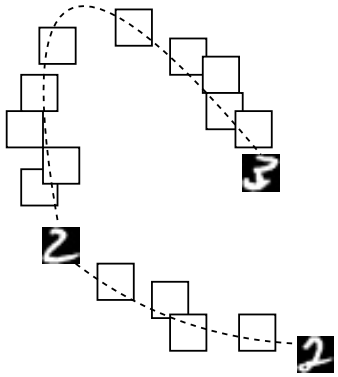


Figure 2.3. Computing geodesic distances between a pair of points on the manifold as a dissimilarity measure.

property to represent a set of distributions or histograms. Below is such an example from [2].

Example 2.2.2. See Fig. 2.4. "We consider 30 images of nested ellipses on a 100×100 grid. Each image is a discrete measure on $[0, 1]^2$ with normalized intensities. Computing the Euclidean, Gaussian RKHS mean-maps or Jeffrey centroid of these images results in mean measures that hardly make any sense, whereas the 2-Wasserstein mean captures perfectly the structure of these images." [2]

2.2.2 Document and Sentence Modeling

In the previous section, we have discussed the motivation of using OT for image comparison, with a particular emphasis on its application to low dimensional histogram data. In this section, we posit that OT can also be a powerful tool for high dimensional discrete distributions.

Our example is in the area of document and sentence modeling. In this area, a common idea to represent a document or a sentence is the bag-of-words (BoW) or bag-of-concepts model. Consider that in a text corpus, we can form a vocabulary of total N unique words and treat each document as a vector of dimension N. Each dimension in the vector denotes the number of times an individual word appears in the document. Therefore, the BoW vector of a document with m words in total at most has m nonzeros, which is a sparse vector. Comparing the similarity between

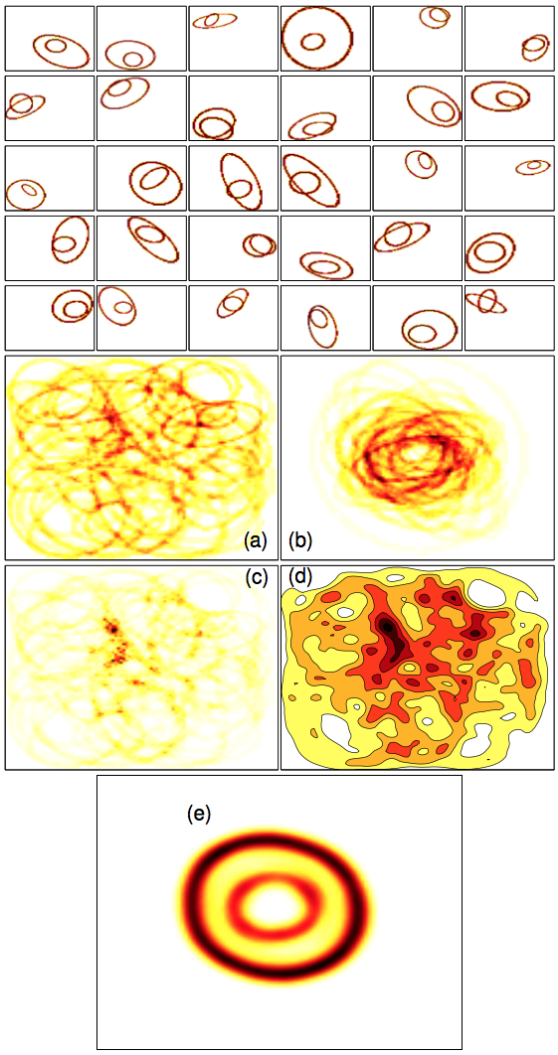


Figure 2.4. (Top) 30 artificial images of two nested random ellipses. Mean measures using the (a) Euclidean distance (b) Euclidean after re-centering images (c) Jeffrey centroid (Nielsen, 2013) (d) RKHS distance (Gaussian kernel, $\sigma = 0.002$) (e) 2-Wasserstein distance. The figure is cropped from [2].

two documents to a large extent is thus to count the occurrences of individuals words across the entire vocabulary by computing the inner product between two sparse BoW vectors.

One assumption of this approach is that documents that share more words have a higher possibility of relatedness or proximity. However, this assumption can be challenged by creating an example like follows.

Example 2.2.3. Consider a sample sentence from the Internet:

- NASA reveals its ambitions that humans can set foot on Mars.

Which one of the three below is believed to tell about a different thing of the sample sentence?

1. United States is planning to send American astronauts to Red Planet.
2. Challenge for the pioneering missions is keeping the crew safe to the destination, said NASA.
3. Electricity is to motor as ambition is to human.

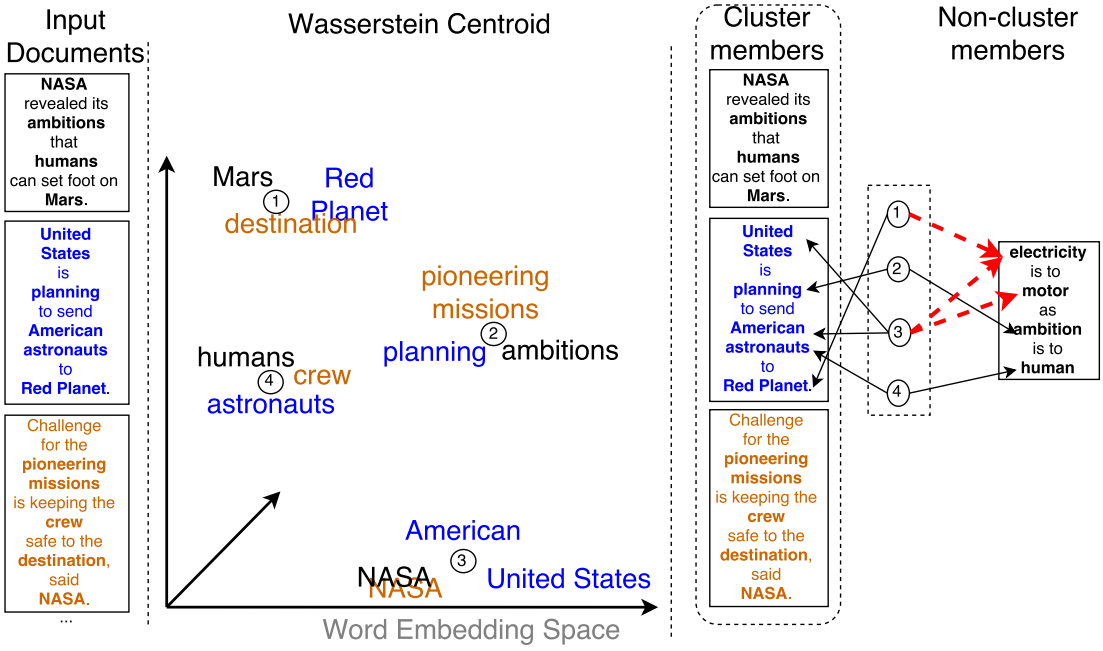


Figure 2.5. Graphical representation of sentences in a word embedding space. The Wasserstein barycenter/centroid of three sentences can capture their characteristic composition with optimal transport matching.

It is clear that the last sentence of the three is telling a different thing. But if we count how many co-occurred non-stop words are shared between the sample sentence and the last sentence, we would easily spot two: "human" and "ambition", which is more than the other two candidate sentences. A common strategy to address the above scenario is to embed all words into a Euclidean space capturing

the semantics of each word. Two words sharing similar semantics would sit in nearby locations in the embedding space. By the embedding technique, we can instead treat a sentence as a point cloud or a discrete distribution. Measuring the dissimilarity between two sentences becomes measuring the distance between two discrete distributions. Wasserstein distance is a powerful tool for measuring semantic dissimilarity between two sentences by principally considering the cross-word semantic similarity captured by the word embedding vectors. Fig. 2.5 depicts how Wasserstein space characterizes the compositional structure of sentences. In particular, we can compute the Wasserstein barycenter/centroid (a concept to be defined in Chapter 4) to summarize the shared composition.

Chapter 3
Computational Optimal Transport: A Cookbook of Numerical Methods

3.1 Introduction

An oracle is a computational module in an optimization procedure that is applied iteratively to obtain certain characteristics of the function being optimized. Typically, it calculates the value and gradient of loss function $l(\mathbf{x}, \mathbf{y})$. In the vast majority of machine learning models, where those loss functions are decomposable along each dimension (*e.g.*, L_p norm, KL divergence, or hinge loss), $\nabla_{\mathbf{x}} l(\cdot, \mathbf{y})$ or $\nabla_{\mathbf{y}} l(\mathbf{x}, \cdot)$ is computed in $O(m)$ time, m being the complexity of outcome variables $\mathbf{x}$ or $\mathbf{y}$. This part of calculation is often negligible compared with the calculation of full gradient with respect to the model parameters. But this is no longer the case in learning problems based on Wasserstein distance due to the intrinsic complexity of the distance. We will call such problems *Wasserstein loss minimization* (WLM). Examples of WLMs include Wasserstein barycenters [2, 8, 11, 19–21], principal geodesics [22], nonnegative matrix factorization [23, 24], barycentric coordinate [25], and multi-label classification [26].

Wasserstein distance (See Eq. (2.1)) is defined as the cost of matching two probability measures, originated from the literature of optimal transport (OT) [27]. It takes into account the cross-term similarity between different support points of

the distributions, a level of complexity beyond the usual vector data treatment, *i.e.*, to convert the distribution into a vector of frequencies. It has been promoted for comparing sets of vectors (*e.g.* bag-of-words models) by researchers in computer vision, multimedia and more recently natural language processing [5, 9]. However, its potential as a powerful loss function for machine learning has been underexplored. The major obstacle is a lack of standardized and robust numerical methods to solve WLMs. Even to empirically better understand the advantages of the distance is of interest.

This section will present three numerical approaches for approximately solving optimal transport problem. They can be used as build blocks for algorithms which computationally tackle more sophisticated problems formulated by OT, which will be visited in Chapter 4.

3.2 Problem Formulation

In this section, we present notations, mathematical backgrounds, and set up the problem of interest.

Definition 3.2.1 (Optimal Transportation, OT)**.** Let $\mathbf{p} \in \Delta_{m_1}, \mathbf{q} \in \Delta_{m_2}$, where Δ_m is the set of m-dimensional simplex: $\Delta_m \stackrel{\text{def.}}{=} \{\mathbf{q} \in \mathbb{R}_+^m : \langle \mathbf{q}, \mathbb{1} \rangle = 1\}$. The set of transportation plans between $\mathbf{p}$ and $\mathbf{q}$ is defined as $\Pi(\mathbf{p}, \mathbf{q}) \stackrel{\text{def.}}{=} \{Z \in \mathbb{R}^{m_1 \times m_2} : Z \cdot \mathbb{1}_{m_2} = \mathbf{p}; Z^T \cdot \mathbb{1}_{m_1} = \mathbf{q}; \}$. Let $M \in \mathbb{R}_+^{m_1 \times m_2}$ be the matrix of costs. The optimal transport cost between $\mathbf{p}$ and $\mathbf{q}$ with respect to M is

$$W(\mathbf{p}, \mathbf{q}) \stackrel{\text{def.}}{=} \min_{Z \in \Pi(\mathbf{p}, \mathbf{q})} \langle Z, M \rangle \ . \tag{3.1}$$

In particular, $\Pi(\cdot, \cdot)$ is often called the coupling set.

Now we relate primal version of (discrete) OT to a variant of its dual version. One may refer to [1] for the general background of the Kantorovich-Rubenstein duality. In particular, our formulation introduces an auxiliary parameter C_M for the sake of mathematical soundness in defining Boltzmann distributions.

Definition 3.2.2 (Dual Formulation of OT)**.** Let $C_M > 0$, denote vector $[g_1, \ldots, g_{m_1}]^T$ by $\mathbf{g}$, and vector $[h_1, \ldots, h_{m_2}]^T$ by $\mathbf{h}$. We define the dual domain of OT by

$$\Omega(M) \stackrel{\text{def.}}{=} \Big\{ \mathbf{f} = [\mathbf{g}; \mathbf{h}] \in \mathbb{R}^{m_1+m_2} \;\Big|\; -C_M < g_i - h_j \le M_{i,j}, 1 \le i \le m_1, 1 \le j \le m_2 \Big\} . \quad (3.2)$$

Informally, for a sufficiently large C_M (subject to $\mathbf{p}, \mathbf{q}, M$), the LP problem Eq. (3.1) can be reformulated as [1]

$$W(\mathbf{p}, \mathbf{q}) = \max_{\mathbf{f} \in \Omega(M)} \langle \mathbf{p}, \mathbf{g} \rangle - \langle \mathbf{q}, \mathbf{h} \rangle . \quad (3.3)$$

Let the optimum set be $\Omega^*(M)$. Then any optimal point $\mathbf{f}^* = (\mathbf{g}^*, \mathbf{h}^*) \in \Omega^*(M)$ constructs a (projected) subgradient such that $\mathbf{g}^* \in \partial W/\partial \mathbf{p}$ and $-\mathbf{h}^* \in \partial W/\partial \mathbf{q}$. The main computational difficulty of WLMs comes from the fact that (projected) subgradient $\mathbf{f}^*$ is not efficiently solvable.

Note that $\Omega(M)$ is an unbound set in $\mathbb{R}^{m_1+m_2}$. In order to constrain the feasible region to be bounded, we alternatively define

$$\Omega_0(M) = \{\mathbf{f} = [\mathbf{g}; \mathbf{h}] \in \Omega(M) \mid g_1 = 0\}. \quad (3.4)$$

One can show that the maximization in $\Omega(M)$ as Eq. (3.3) is equivalent to the maximization in $\Omega_0(M)$ because $\langle \mathbf{p}, \mathbb{1}_{m_1} \rangle = \langle \mathbf{q}, \mathbb{1}_{m_2} \rangle$.

3.3 Sinkhorn-Knopp Algorithm and Bregman Alternating Direction Method of Multiplier

In this section, we will describe two approaches that approximately solve optimal transport in near-linear time. These two approaches have been used to solve more sophisticated Wasserstein problems, which will be introduced in later chapters in this thesis.

Sinkhorn-Knopp Algorithm. Cuturi [28] introduced a smoothed approach to approximate the original OT and proposed the use of the Sinkhorn-Knopp algorithm to solve the smoothed problem. In its formulation, the original OT Eq. (3.1) is

[1] However, for any proper M and strictly positive $\mathbf{p}, \mathbf{q}$, there exists C_M such that the optimal value of primal problem is equal to the optimal value of the dual problem. This modification is solely for an ad-hoc treatment of a single OT problem. In general cases of $(\mathbf{p}, \mathbf{q}, M)$, when C_M is pre-fixed, the solution of Eq. (3.3) may be suboptimal.

instead approximated by the following optimization: For some $\eta > 0$,

$$\min_{Z \in \Pi(\mathbf{p},\mathbf{q})} \langle Z, M \rangle - \eta^{-1} H(Z), \tag{3.5}$$

where $H(Z)$ is the entropy of joint probability Z. This is called entropic regularization in the literature. Cuturi [28] shows that one can approximately solve the modified problem up to an error tolerance ε' by call SINKHORN$(\exp(\eta M), \Pi(\mathbf{p}, \mathbf{q}), \varepsilon')$ (See Algorithm 3.1). In brief, it is an alternating projection procedure which renormalizes the rows and columns of A in turn so that they match the desired row and column marginals r and c.

Algorithm 3.1 Sinkhorn algorithm for OT

procedure SINKHORN(A, $\Pi_{r,c}$, ε')
  initialize $k = 0$
  $A^{(0)} \leftarrow A/\|A\|_1$, $\mathbf{x}^0 \leftarrow \mathbf{0}$, $\mathbf{y}^0 \leftarrow \mathbf{0}$.
  while dist$(A^{(k)}, \Pi_{r,c}) < \varepsilon'$ **do**
    $k \leftarrow k + 1$
    if k odd **then**
      $x_i \leftarrow \log\left(\frac{r_i}{r_i(A^{(k-1)})}\right)$ for $i \in [m_1]$ $\#r_i(\cdot)$ is the sum of i-th row
      $\mathbf{x}^k \leftarrow \mathbf{x}^{k-1} + \mathbf{x}$, $\mathbf{y}^k \leftarrow \mathbf{y}^{k-1}$
    else
      $y_i \leftarrow \log\left(\frac{c_i}{c_i(A^{(k-1)})}\right)$ for $i \in [m_2]$ $\#c_i(\cdot)$ is the sum of i-th column
      $\mathbf{y}^k \leftarrow \mathbf{y}^{k-1} + \mathbf{y}$, $\mathbf{x}^k \leftarrow \mathbf{x}^{k-1}$
    end if
    $A^{(k)} = D(\exp(\mathbf{x}^k))AD(\exp(\mathbf{y}^k))$ $\#D(\cdot)$ is the diagonal matrix from vector
  end while
  Output $B \leftarrow A^{(k)}$
end procedure

Let $A = \exp(\eta M)$. If one measure the dist$(A, \Pi_{r,c})$ by $\|r(A) - r\|_2 + \|c(A) - c\|_2$, Kalantari et al. [29] shows SINKHORN outputs a matrix B satisfying dist$(B, \Pi_{r,c}) < \varepsilon$ in $O\left(\rho(\varepsilon')^{-2} \log(s/l)\right)$ iterations, with $s = \sum_{ij} A_{ij}$ and $l = min_{ij} A_{ij}$ and $\rho \geq r_i, c_j$ for all i and j. However, as mentioned by [30], ℓ_2 is not an appropriate measure for simplex constrained probabilities. Adapting the results to ℓ_1, we must take $O\left(\max\{m_1, m_2\}\rho(\varepsilon')^{-2} log(s/l)\right)$ iterations of SINKHORN to output B such that $\|r(A) - r\|_1 + \|c(A) - c\|_1 < \varepsilon'$. The extra factor of $\max\{m_1, m_2\}$ is the price of converting an ℓ_2 bound to an ℓ_1 bound. Altschuler et al. [30] recently gave a new analysis for SINKHORN, showing we can obtain an ℓ_1 bound in $O\left((\varepsilon')^{-2} \log(s/l)\right)$

iterations. He also showed the solution $\hat{Z}$ obtained via Sinkhorn algorithm satisfies

$$\langle \hat{Z}, M \rangle \leq \min_{Z \in \Pi(\mathbf{p},\mathbf{q})} \langle Z, M \rangle + \frac{2 \log n}{\eta} + 4\varepsilon' \|M\|_\infty,$$

where $n = \max\{m_1, m_2\}$. Finally, let $\eta = \frac{4 \log n}{\varepsilon}$ and $\varepsilon' = \frac{\varepsilon}{8\|M\|_\infty}$, SINKHORN outputs an approximate solution $\hat{Z}$ with ε guarantee in the objective, ε' ℓ_1-guarantee in constraints in $O\left((\varepsilon')^{-2}(\log n + \eta \|M\|_\infty)\right)$ Sinkhorn iterations, a.k.a. $O\left(\frac{m_1 m_2 \log \max\{m_1, m_2\} \|M\|_\infty^3}{\varepsilon^3}\right)$ time. Remember each Sinkhorn iteration takes $O(m_1 m_2)$ time.

The analysis of [30] answered a long-standing problem in this line of research, in which entropic regularization is used to smooth the classic OT. For computing two distributions with large support size, Sinkhorn is a favorable technique because it is proved to have near-linear performance with respect to the support size. However, the convergence to the true problem is still quite slow for small ε, as it takes roughly $O(\varepsilon^{-3})$ iterations. This probably explains why researchers have used the smoothed solution from entropic regularization for its own right.

To approximate the classic OT using Sinkhorn iterations, we must address two issues. One is the slow convergence rate with respect to the error bound. The other is that the Sinkhorn algorithm can easily run out of float precisions in practice. To complement these two issues, we may instead consider a different technique, which will be introduced next.

It is quite convenient to recover an approximate dual solution from the outputs of Sinkhorn algorithm. The entropic regularized problem is in fact can be reformulated via a dual perspective [3]:

$$\max_{\mathbf{g},\mathbf{h}} \langle \mathbf{p}, \mathbf{g} \rangle - \langle \mathbf{q}, \mathbf{h} \rangle - \eta^{-1} \sum_{i,j} \exp\{-\eta \left(M_{i,j} - g_i + h_j\right)\} \tag{3.6}$$

Let $\mathbf{x}^T, \mathbf{y}^T$ be the outputs variables updated in Sinkhorn algorithm. Then the $\mathbf{g}^* \approx -\eta^{-1}\mathbf{x}$, $\mathbf{h}^* \approx \eta^{-1}\mathbf{y}$ are the dual solution [2, 3].

Bregman ADMM. Wang and Banerjee [31] introduced a variant of alternative direction method of multiplier (ADMM) to solve optimal transport problem with

provable guarantees. The general idea is to first decouple constraint set $\Pi(\mathbf{p}, \mathbf{q})$ by rewriting the optimization problem as

$$\min_{\substack{r(Z_1)=\mathbf{p} \\ c(Z_2)=\mathbf{q}}} \langle Z_1, M \rangle \text{ s.t. } Z_1 = Z_2. \tag{3.7}$$

Bregman ADMM updates proceed by the solving the following proxy subproblems.

$$\begin{aligned} Z_1 &:= \arg\min_{r(Z_1)=\mathbf{p}} \langle Z_1, M \rangle + \langle \Lambda, Z_1 \rangle + \underbrace{\rho \cdot \mathrm{KL}(Z_1, Z_2)}_{\text{replace } |\cdot|^2 \text{ with } B_\Phi(\cdot,\cdot)} \\ Z_2 &:= \arg\min_{c(Z_2)=\mathbf{q}} -\langle \Lambda, Z_2 \rangle + \rho \cdot \mathrm{KL}(Z_2, Z_1) \\ \Lambda &:= \Lambda + \rho(Z_1 - Z_2) \end{aligned}$$

Note that Bregman ADMM replaces the regular ℓ_2 penalty with the Bregman divergence. Each subproblems are in fact solvable in closed form. Hence the B-ADMM approach allows a new algorithm for classic OT (See Algorithm 3.2)

Algorithm 3.2 Bregman ADMM for OT (See Matlab notations)

procedure B-ADMM(M, $\Pi(\mathbf{p}, \mathbf{q})$, ρ)
  Initialize $Z_2^0 = \mathbf{p}\mathbf{q}^T$
  for $t = 1, \ldots, T$ **do**
    $Z_1 \leftarrow Z_2^{t-1} \odot \exp\left\{-\dfrac{M + \Lambda^{t-1}}{\rho}\right\}$ #$\odot$ is element-wise product.
    $Z_1^t \leftarrow \texttt{bsxfun}(\texttt{@times}, Z_1, \mathbf{p}./r(Z_1))$
    $Z_2 \leftarrow Z_1^t \odot \exp\left\{\dfrac{\Lambda^{t-1}}{\rho}\right\}$
    $Z_2^t \leftarrow \texttt{bsxfun}(\texttt{@times}, Z_2, \mathbf{q}^T./c(Z_2))$
    $\Lambda^t \leftarrow \Lambda^{t-1} + Z_1^t - Z_2^t$
  end for
  Output $\bar{Z}_1 = \dfrac{Z_1^1 + \ldots + Z_1^T}{T}$
end procedure

Wang and Banerjee [31] gave a general analysis for Bregman ADMM method. We adapt their results to optimal transport and have the following convergence guarantees: Suppose the optimal transport have KKT a solution $W^* = (Z^*, \Lambda^*)$,

	Objective	Constraint ℓ_1
Sinkhorn	$\tilde{O}(T^{-\frac{1}{3}})$	$\tilde{O}(T^{-\frac{2}{3}})$
B-ADMM	$\tilde{O}(\rho T^{-1})$	$\tilde{O}(\sqrt{m_1 m_2}\rho^{-1}T^{-\frac{1}{2}})$

Table 3.1. Convergence rate for T iterations.

we let

$$D(W^*, W^t) = \mathrm{KL}(Z^*, Z_2^t) + \frac{1}{\rho^2}\|\Lambda^* - \Lambda^t\|^2,$$

then we have

$$\mathrm{KL}(Z_1^{t+1}, Z_2^t) \leq \underbrace{D(W^*, W^t)}_{\text{monotonic nonincreasing}} - D(W^*, W^{t+1}).$$

Moreover it also presents guaranteed optimality:

$$\langle \bar{Z}_1^T, M\rangle - \langle Z^*, M\rangle \leq \frac{\rho \mathrm{KL}(Z^*, Z_2^0)}{T},$$

$$\|\bar{Z}_1^T - \bar{Z}_2^T\|_2 \leq \sqrt{\frac{2D(W^*, W^0)}{T}},$$

where $\bar{Z_j}^T = \frac{1}{T}\sum_{t=1}^T Z_j^t$, $j = 1, 2$. Let $n = \max\{m_1, m_2\}$ Since $D(W^*, W^0) \leq 2\log n + \|\Lambda^*\|_2^2\rho^{-2}$ and $\mathrm{KL}(Z^*, Z_2^0) \leq 2\log n$, we let $T = 2\rho\log n\varepsilon^{-1}$, B-ADMM can reach an approximate solution with ε guarantee in objective and

$$\sqrt{m_1 m_2}\left(\rho^{-1} + \frac{\|\Lambda^*\|_2^2}{2\log n}\rho^{-3}\right)^{\frac{1}{2}} \cdot \varepsilon^{-\frac{1}{2}}$$

ℓ_1-guarantee in constraints. In other words, B-ADMM takes much less number of iterations to generate an approximation to the objective function while takes much more number of iterations to generate an approximation that satisfies a constraint measurement. Table 3.1 compares Sinkhorn and B-ADMM if the same number of iterations are budgeted.

In order to recover an approximate dual solution from the outputs of Bregman ADMM, one may consider the complementary slackness of KKT condition:

$$Z_{ij}(g_i - h_j - M_{ij}) \approx 0 \tag{3.8}$$

By solving a least square problem, one can find the dual solution in linear time (from a preconditioned linear system): for some $\lambda > 0$,

$$\min_{\mathbf{g},\mathbf{h}} \sum_{i,j} Z_{ij}(g_i - h_j - M_{ij})^2 + \lambda\|\mathbf{g}\|^2 + \lambda\|\mathbf{h}\|^2. \tag{3.9}$$

3.4 Simulated Annealing

As a long-standing consensus, solving WLMs is challenging [2]. Unlike the usual optimization in machine learning where the loss and the (partial) gradient can be calculated in linear time, these quantities are non-smooth and hard to obtain in WLMs, requiring solution of a costly network transportation problem (a.k.a. OT). The time complexity, $O(m^3 \log m)$, is prohibitively high [33]. In contrast to the L_p or KL counterparts, this step of calculation elevates from a negligible fraction of the overall learning problem to a dominant portion, preventing the scaling of WLMs to large data. Recently, iterative approximation techniques have been developed to compute the loss and the (partial) gradient at complexity $O(m^2/\varepsilon)$ [28, 31]. However, nontrivial algorithmic efforts are needed to incorporate these methods into WLMs because WLMs often require multi-level loops [2, 26]. Specifically, one must re-calculate through many iterations the loss and its partial gradient in order to update other model dependent parameters.

We are thus motivated to seek for a fast *inexact* oracle that (i) runs at lower time complexity per iteration, and (ii) accommodates warm starts and meaningful early stops. These two properties are equally important for efficiently obtaining adequate approximation to the solutions of a sequence of slowly changing OTs. The second property ensures that the subsequent OTs can effectively leverage the solutions of the earlier OTs so that the total computational time is low. Approximation techniques with low complexity per iteration already exist for solving a single OT, but they do not possess the second property. In this paper, we introduce a method that uses a time-inhomogeneous Gibbs sampler as an inexact oracle for Wasserstein

The work presented in this section has been published in the form of a research paper: Jianbo Ye, James Z. Wang and Jia Li, "A Simulated Annealing based Inexact Oracle for Wasserstein Loss Minimization," *Proceedings of the 34th International Conference on Machine Learning (ICML)*, Sydney, Australia, Vol 70, pp 3940–3948, August 2017. It was also presented in BIRS-CMO Workshop [32].

losses. The Markov chain Monte Carlo (MCMC) based method naturally satisfies the second property, as reflected by the intuition of physicists that MCMC samples can efficiently "remix from a previous equilibrium."

We propose a new optimization approach based on Simulated Annealing (SA) [34, 35] for WLMs where the outcome variables are treated as probability measures. SA is especially suitable for the dual OT problem, where the usual Metropolis sampler can be simplified to a Gibbs sampler. To our knowledge, existing optimization techniques used on WLMs are different from MCMC. In practice, MCMC is known to easily accommodate warm start, which is particularly useful in the context of WLMs. We name this approach *Gibbs-OT* for short. The algorithm of Gibbs-OT is as simple and efficient as the Sinkhorn's algorithm — a widely accepted method to approximately solve OT [28]. We show that Gibbs-OT enjoys improved numerical stability and several algorithmic characteristics valuable for general WLMs. By experiments, we demonstrate the effectiveness of Gibbs-OT for solving optimal transport with Coulomb cost [36] and the Wasserstein non-negative matrix factorization (NMF) problem [23, 24].

3.4.1 Optimal Transport via Gibbs Sampling

Following the basic strategy outlined in the seminal paper of simulated annealing [34], we present the definition of Boltzmann distribution supported on $\Omega_0(M)$ below which, as we will elaborate, links the dual formulation of OT to a Gibbs sampling scheme (Algorithm 3.3 below).

Definition 3.4.1 (Boltzmann Distribution of OT)**.** Given a temperature parameter $T > 0$, the Boltzmann distribution of OT is a probability measure on $\Omega_0(M) \subseteq \mathbb{R}^{m_1+m_2-1}$ such that

$$p(\mathbf{f}; \mathbf{p}, \mathbf{q}) \propto \exp\left[\frac{1}{T}\left(\langle \mathbf{p}, \mathbf{g}\rangle - \langle \mathbf{q}, \mathbf{h}\rangle\right)\right]. \tag{3.10}$$

It is a well-defined probability measure for an arbitrary finite $C_M > 0$.

The basic concept behind SA states that the samples from the Boltzmann distribution will eventually concentrate at the optimum set of its deriving problem (*e.g.* $W(\mathbf{p}, \mathbf{q})$) as $T \to 0$. However, since the Boltzmann distribution is often difficult

to sample, a practical convergence rate remains mostly unsettled for specific MCMC methods.

Because $\Omega(M)$ defined by Eq. (3.2) (also Ω_0) has a conditional independence structure among variables, a Gibbs sampler can be naturally applied to the Boltzmann distribution defined by Eq. (3.10). We summarize this result below.

Proposition 3.4.1. *Given any* $\mathbf{f} = (\mathbf{g}; \mathbf{h}) \in \Omega_0(M)$ *and any* $C_M > 0$*, we have for any* i *and* j*,*

$$g_i \le U_i(\mathbf{h}) \overset{\text{def.}}{=} \min_{1 \le j \le m_2} (M_{i,j} + h_j), \tag{3.11}$$

$$h_j \ge L_j(\mathbf{g}) \overset{\text{def.}}{=} \max_{1 \le i \le m_1} (g_i - M_{i,j}). \tag{3.12}$$

and

$$g_i > \widehat{L}_i(\mathbf{h}) \overset{\text{def.}}{=} \max_{1 \le j \le m_2} (-C_M + h_j), \tag{3.13}$$

$$h_j < \widehat{U}_j(\mathbf{g}) \overset{\text{def.}}{=} \max_{1 \le i \le m_1} (C_M + g_i). \tag{3.14}$$

Here $U_i = U_i(\mathbf{h})$ *and* $L_j = L_j(\mathbf{g})$ *are auxiliary variables. Suppose* $\mathbf{f}$ *follows the Boltzmann distribution by Eq. (3.10),* g_i*'s are conditionally independent given* $\mathbf{h}$*, and likewise* h_j*'s are also conditionally independent given* $\mathbf{g}$*. Furthermore, it is immediate from Eq. (3.10) that each of their conditional probabilities within its feasible region (subject to* C_M*) satisfies*

$$p(g_i|\mathbf{h}) \propto \exp\left(\frac{g_i p_i}{T}\right), \ \widehat{L}_i(\mathbf{h}) < g_i \le U_i(\mathbf{h}), \tag{3.15}$$

$$p(h_j|\mathbf{g}) \propto \exp\left(-\frac{h_j q_j}{T}\right), \ L_j(\mathbf{g}) \le h_j < \widehat{U}_j(\mathbf{g}), \tag{3.16}$$

where $2 \le i \le m_1$ *and* $1 \le j \le m_2$*.*

Remark 2. As $C_M \to +\infty$, $\widehat{U}_j(\mathbf{g}) \to +\infty$ and $\widehat{L}_i(\mathbf{h}) \to -\infty$. For $2 \le i \le m_1$ and $1 \le j \le m_2$, one can approximate the conditional probability $p(g_i|\mathbf{h})$ and $p(h_j|\mathbf{g})$ by exponential distributions.

By Proposition. 3.4.1, our proposed time-inhomogeneous Gibbs sampler is given in Algorithm 3.3. Specifically in Algorithm 3.3, the variable g_1 is fixed to zero by the definition of $\Omega_0(M)$. But we have found in experiments that by calculating $U_1^{(t)}$

Algorithm 3.3 Gibbs Sampling for Optimal Transport

Given $\mathbf{f}^{(0)} \in \Omega_0(M)$, $\mathbf{p} \in \Delta_{m_1}$ and $\mathbf{q} \in \Delta_{m_2}$, and $T^{(1)}, \ldots, T^{(2N)} > 0$, for $t = 1, \ldots, N$, we define the following Markov chain

1. Randomly sample

$$\theta_1, \ldots, \theta_{m_2} \overset{i.i.d.}{\sim} \text{Exponential}(1).$$

For $j = 1, 2, \ldots, m_2$, let

$$\begin{cases} L_j^{(t)} := & \max_{1 \le i \le m_1} \left(g_i^{(t-1)} - M_{i,j} \right) \\ h_j^{(t)} := & L_j^{(t)} + \theta_j \cdot T^{(2t-1)} / q_j \end{cases} \tag{3.17}$$

2. Randomly sample

$$\theta_2, \ldots, \theta_{m_1} \overset{i.i.d.}{\sim} \text{Exponential}(1).$$

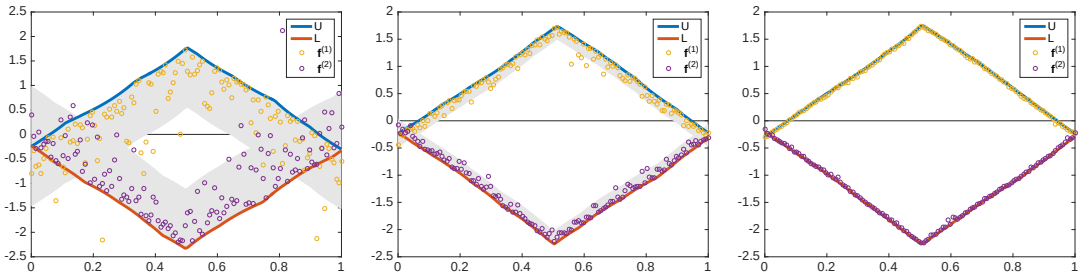
For $i = (1), 2, \ldots, m_1$, let

$$\begin{cases} U_i^{(t)} := & \min_{1 \le j \le m_2} \left(M_{i,j} + h_j^{(t)} \right) \\ g_i^{(t)} := & U_i^{(t)} - \theta_i \cdot T^{(2t)} / p_i \end{cases} \tag{3.18}$$

and sampling $g_1^{(t)}$ in Algorithm 3.3 according to Eq. (3.18), one can still generate MCMC samples from $\Omega(M)$ such that the energy quantity $\langle \mathbf{p}, \mathbf{g} \rangle - \langle \mathbf{q}, \mathbf{h} \rangle$ converges to the same distribution as that of MCMC samples from $\Omega_0(M)$. Therefore, we will not assume $g_1 = 0$ from now on and develop analysis solely for the unconstrained version of Gibbs-OT.

Fig. 3.1 illustrates the behavior of the proposed Gibbs sampler with a cooling schedule at different temperatures. As T decreases along iterations, the 95% percentile band for sample $\mathbf{f}$ becomes thinner and thinner.

Remark 3. Algorithm 3.3 does not specify the actual cooling schedule, nor does the analysis of the proposed Gibbs sampler in Theorem 3.4.2. We have been agnostic here for a reason. In the SA literature, cooling schedules with guaranteed optimality are often too slow to be useful in practice. To our knowledge, the guaranteed rate of SA approach is worse than the combinatorial solver for OT. As a result, a well-accepted practice of SA for many complicated optimization problems is to empirically adjust cooling schedules, a strategy we take for our experiments.



(a) 20 iterations (b) 40 iterations (c) 60 iterations

Figure 3.1. The Gibbs sampling of the proposed SA method. From left to right is an illustrative example of a simple 1D optimal transportation problem with Coulomb cost and plots of variables for solving this problem at different number of iterations $\in \{20, 40, 60\}$ using the inhomogeneous Gibbs sampler. Particularly, the 95% percentile of the exponential distributions are marked by the gray area.

Remark 4. Although the exact cooling schedule is not specified, we still provide a quantitative upper bound of the chosen temperature T at different iterations in Sec. 3.4.3 Eq. (3.25). One can calculate such bound at the cost of $m \log m$ at certain iterations to check whether the current temperature is too high for the used Gibbs-OT to accurately approximate the Wasserstein gradient. In practice, we find this bound helps one quickly select the beginning temperature of Gibbs-OT algorithm.

Definition 3.4.2 (Notations for Auxiliary Statistics)**.** Besides the Gibbs coordinates $\mathbf{g}$ and $\mathbf{h}$, the Gibbs-OT sampler naturally introduces two auxiliary variables, $\mathbf{U}$ and $\mathbf{L}$. Let $\mathbf{L}^{(t)} = \left[L_1^{(t)}, \ldots, L_{m_2}^{(t)}\right]^T$ and $\mathbf{U}^{(t)} = \left[U_1^{(t)}, \ldots, U_{m_1}^{(t)}\right]^T$. Likewise, denote the collection of $g_i^{(t)}$ and $h_j^{(t)}$ by vectors $\mathbf{g}^{(t)}$ and $\mathbf{h}^{(t)}$ respectively. The following sequence of auxiliary statistics

$$[\ldots, \mathbf{z}^{2t-1}, \mathbf{z}^{2t}, \mathbf{z}^{2t+1}, \ldots,] \quad \stackrel{\text{def.}}{=} \quad \left[\ldots, \begin{bmatrix} \mathbf{L}^{(t)} \\ \mathbf{U}^{(t-1)} \end{bmatrix}, \begin{bmatrix} \mathbf{L}^{(t)} \\ \mathbf{U}^{(t)} \end{bmatrix}, \begin{bmatrix} \mathbf{L}^{(t+1)} \\ \mathbf{U}^{(t)} \end{bmatrix}, \ldots\right] \tag{3.19}$$

for $t = 1, \ldots, N$ is also a Markov chain. They can be redefined equivalently by specifying the transition probabilities $p(\mathbf{z}^{n+1}|\mathbf{z}^n)$ for $n = 1, \ldots, 2N$, a.k.a., the conditional p.d.f. $p(\mathbf{U}^{(t)}|\mathbf{L}^{(t)})$ for $t = 1, \ldots, N$ and $p(\mathbf{L}^{(t+1)}|\mathbf{U}^{(t)})$ for $t = 1, \ldots, N-1$.

One may notice that the alternative representation converts the Gibbs sampler to one whose structure is similar to a hidden Markov model, where the $\mathbf{g}, \mathbf{h}$ chain

is conditional independent given the $\mathbf{U}, \mathbf{L}$ chain and has (factored) exponential emission distributions. We will use this equivalent representation in Sec. 3.4.3 and develop analysis based on the $\mathbf{U}, \mathbf{L}$ chain accordingly.

Remark 5. We now consider the function

$$V(\mathbf{x}, \mathbf{y}) \stackrel{\text{def.}}{=} \langle \mathbf{p}, \mathbf{x} \rangle - \langle \mathbf{q}, \mathbf{y} \rangle \ ,$$

and define a few additional notations. Let $V(\mathbf{U}^{t'}, \mathbf{L}^{t})$ be denoted by $V(\mathbf{z}^{t+t'})$, where $t' = t$ or $t-1$. If $\mathbf{g}, \mathbf{h}$ are independently resampled according to Eq. (3.17) and (3.18), we will have the inequalities that

$$\mathbb{E}\left[V(\mathbf{g}, \mathbf{h}) | \mathbf{z}^n\right] \leq V(\mathbf{z}^n) \ .$$

Both $V(\mathbf{z})$ and $V(\mathbf{g}, \mathbf{h})$ converges to the exact loss $W(\mathbf{p}, \mathbf{q})$ at the equilibrium of Boltzmann distribution $p(\mathbf{f}; \mathbf{p}, \mathbf{q})$ as $T \to 0$. [2]

3.4.2 Gibbs-OT: An Inexact Oracle for WLMs

In this section, we introduce a non-standard SA approach for the general WLM problems. The main idea is to replace the standard Boltzmann energy with an asymptotic consistent upper bound, outlined in our previous section. Let

$$\mathfrak{R}(\theta) := \sum_{i=1}^{|\mathcal{D}|} W(\mathbf{p}_i(\theta), \mathbf{q}_i(\theta))$$

be our prototyped objective function, where $\mathcal{D}$ represents a dataset, $\mathbf{p}_i, \mathbf{q}_i$ are prototyped probability densities for representing the i-th instance. We now discuss how to solve $\min_{\theta \in \Theta} \mathfrak{R}(\theta)$.

To minimize the Wasserstein losses $W(\mathbf{p}, \mathbf{q})$ approximately in such WLMs, we propose to instead optimize its asymptotic consistent upper bound $\mathbb{E}[V(\mathbf{z})]$ at equilibrium of Boltzmann distribution $p(\mathbf{f}; \mathbf{p}, \mathbf{q})$ using its stochastic gradients: $\mathbf{U} \in \partial V(\mathbf{z})/\partial \mathbf{p}$ and $-\mathbf{L} \in \partial V(\mathbf{z})/\partial \mathbf{q}$. Therefore, one can calculate the gradient

[2] The conditional quantity $V(\mathbf{z}^n) - V(\mathbf{g}, \mathbf{h}) | \mathbf{z}^n$ is the sum of two Gamma random variables: $\text{Gamma}(m_1, 1/T^{(2t)}) + \text{Gamma}(m_2, 1/T^{(2t'+1)})$ where $t' = t$ or $t' = t-1$.

approximately:

$$\nabla_\theta \mathfrak{R} \approx \sum_{i=1}^{|\mathcal{D}|} [J_\theta(\mathbf{p}_i(\theta))\mathbf{U}_i - J_\theta(\mathbf{q}_i(\theta))\mathbf{L}_i]$$

where $J_\theta(\cdot)$ is the Jacobian, $\mathbf{U}_i$, $\mathbf{L}_i$ are computed from Algorithm 3.3 for the problem $W(\mathbf{p}_i, \mathbf{q}_i)$ respectively. Together with the iterative updates of model parameters θ, one gradually anneals the temperature T. The equilibrium of $p(\mathbf{f}; \mathbf{p}, \mathbf{q})$ becomes more and more concentrated. We assume the inexact oracle at a relatively higher temperature is adequate for early updates of the model parameters, but sooner or later it becomes necessary to set T smaller to better approximate the exact loss.

It is well known that the variance of stochastic gradient usually affects the rate of convergence. The reason to replace $V(\mathbf{g}, \mathbf{h})$ with $V(\mathbf{z})$ as the inexact oracle (for some $T > 0$) is motivated by the same intuition. The variances of MCMC samples $g_i^{(t)}, h_j^{(t)}$ of Algorithm 3.3 can be very large if p_i/T and q_j/T are small, making the embedded first-order method inaccurate unavoidably. But we find the variances of max/min statistics $U_i^{(t)}, L_j^{(t)}$ are much smaller. Fig. 3.1 shows an example. The bias introduced in the replacement is also well controlled by decreasing the temperature parameter T. For the sake of efficiency, we use a very simple convergence diagnostics in the practice of Gibbs-OT. We check the values of $V(\mathbf{z}^{(2t)})$ such that the Markov chain is roughly considered mixed if every τ iteration the quantity $V(\mathbf{z}^{(2t)})$ (almost) stops increasing (τ=5 by default), say, for some t,

$$V(\mathbf{z}^{(2t)}) - V(\mathbf{z}^{(2(t-\tau))}) < 0.01\tau T \cdot V(\mathbf{z}^{(2t)}),$$

we terminate the Gibbs iterations.

3.4.3 Theoretical Properties of Gibbs-OT

We develop quantitative concentration bounds for Gibbs-OT in a finite number of iterations in order to understand the relationship between the temperature schedule and the concentration progress. The analysis also guides us to adjust cooling schedule on-the-fly, as will be shown. Proofs are provided in Supplement.

Preliminaries. Before characterizing the properties of Gibbs-OT by Definition 3.3, we first give the analytic expression for $p(\mathbf{z}^{n+1}|\mathbf{z}^n)$. Let $G(\cdot) : [-\infty, \infty] \mapsto [0, 1]$ be the c.d.f. of standard exponential distribution. Because $L_j^{(t+1)} < x$ by definition

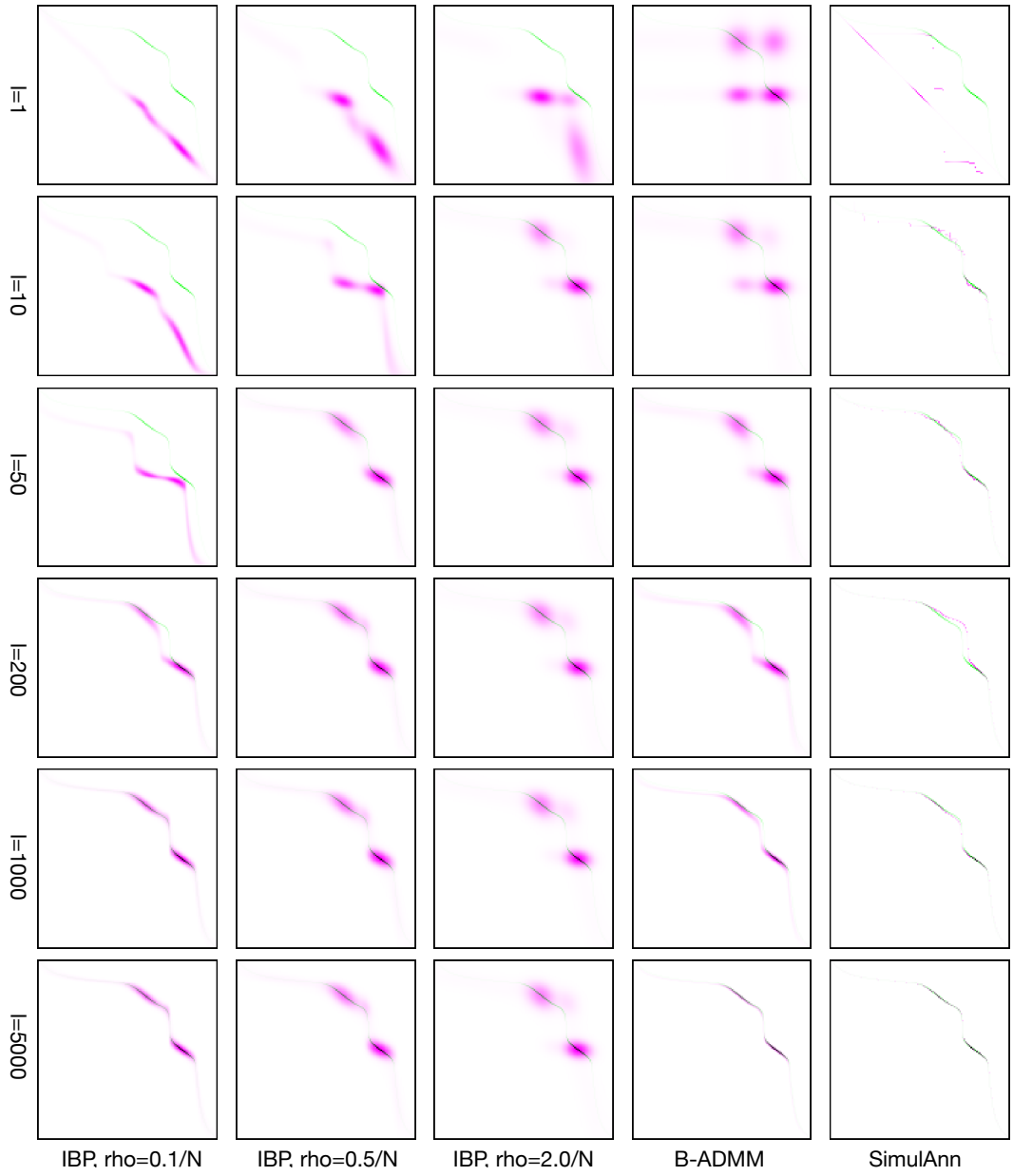


Figure 3.2. A simple example for OT between two 1D distribution: The solutions by Iterative Bregman Projection, B-ADMM, and Gibbs-OT are shown in pink, while the exact solution by linear programming is shown in green. Images in the rows from top to bottom present results at different iterations $\{1, 10, 50, 200, 1000, 5000\}$; The left three columns are by IBP with $\varepsilon = \{0.1/N, 0.5/N, 2/N\}$, where $[0, 1]$ is discretized with $N = 128$ uniformly spaced points. The fourth column is by B-ADMM (with default parameter $\tau_0 = 2.0$). The last column is the proposed Gibbs-OT, with a geometric cooling schedule. With a properly selected cooling schedule, one can achieve fast convergence of OT solution without comprising much solution quality.

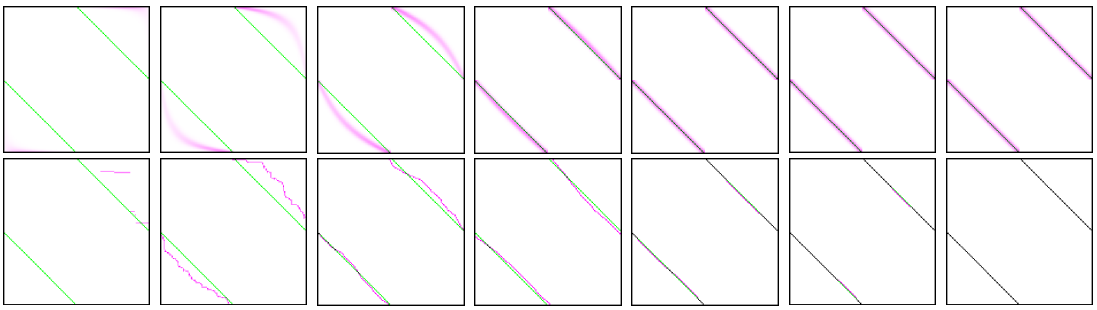


Figure 3.3. The recovered primal solutions for two uniform 1D distribution with Coulumb cost. The approximate solutions are shown in pink, while the exact solution by linear programming is shown in green. Top row: entropic regularization with $\varepsilon = 0.5/N$. Bottom row: Gibbs-OT. Images in the rows from left to right present results at different max iterations $\{1, 10, 50, 200, 1000, 2000, 5000\}$.

$\Leftrightarrow \forall i, \quad g_i^{(t)} - M_{i,j} < x$, the c.d.f. of $L_j^{(t+1)} | \mathbf{U}^{(t)}$ reads

$$\Pr\left(L_j^{(t+1)} < x \,\middle|\, \mathbf{U}^{(t)}\right) = \prod_{i=1}^{m_1} \left(1 - G\left(\frac{-x - M_{i,j} + U_i^{(t)}}{T^{(2t)}/p_i}\right)\right).$$

Likewise, the c.d.f. of $U_i^{(t)} | \mathbf{L}^{(t)}$ reads

$$\Pr\left(U_i^{(t)} < x \,\middle|\, \mathbf{L}^{(t)}\right) = \prod_{j=1}^{m_2} G\left(\frac{x - M_{i,j} - L_j^{(t)}}{T^{(2t-1)}/q_j}\right).$$

With some calculation, the following can be shown. As a note, this lemma provides an intermediate result whose main purpose is to lay down the definition of $\phi_j^{(t)}$ and $\varphi_i^{(t)}$, which are then used in defining $O(z, T)$ (Eq. (3.22)) and r^n (Eq. (3.24)) and in Theorem 3.4.2.

Lemma 3.4.1. *(i) Given $1 \le j \le m_2$ and $1 \le t \le N$, let the sorted index of $\{U_i^{(t)} - M_{i,j}\}_{i=1}^{m_1}$ be permutation $\{\sigma(i)\}_{i=1}^{m_1}$ such that sequence $\{U_{\sigma(i)}^{(t)} - M_{\sigma(i),j}\}_{i=1}^{m_1}$ are monotonically non-increasing. Define the auxiliary quantity*

$$\phi_j^{(t)} \stackrel{\text{def.}}{=} \sum_{k=1}^{m_1} \frac{(1 - \mu_k) \prod_{i=1}^{k-1} \mu_i}{\sum_{i=1}^{k} p_{\sigma(k)}}, \tag{3.20}$$

where

$$1 \quad \geq \quad \mu_i \quad \stackrel{\text{def.}}{=} \quad \exp\left\{ \frac{\sum_{i=1}^{k} p_{\sigma(k)}}{T^{(2t)}} \left[\left(U_{\sigma(i+1)} - M_{\sigma(i+1),j} \right) \; - \; \left(U_{\sigma(i)} - M_{\sigma(i),j} \right) \right] \right\}$$

for $i = 1, \ldots, m_1 - 1$, *and* $\mu_{m_1} \stackrel{\text{def.}}{=} 0$. *Then, the conditional expectation*

$$\mathbb{E}\left[L_j^{(t+1)} \middle| \mathbf{U}^{(t)} \right] = U_{\sigma(1)}^{(t)} - M_{\sigma(1),j} - \phi_j^{(t)} T^{(2t)} \; .$$

In particular, we denote $\sigma(1)$ *by* I_j^t *or* $I(j,t)$.

(ii) Given $1 \leq i \leq m_1$ *and* $1 \leq t \leq N$, *let the sorted index of* $\{M_{i,j} + L_j\}_{j=1}^{m_2}$ *be permutation* $\{\sigma(j)\}_{j=1}^{m_2}$ *such that the sequence* $\{M_{i,\sigma(j)} + L_{\sigma(j)}^{(t)}\}_{j=1}^{m_2}$ *are monotonically non-decreasing. Define the auxiliary quantity*

$$\psi_i^{(t)} \stackrel{\text{def.}}{=} \sum_{k=1}^{m_2} \frac{(1 - \lambda_k) \prod_{j=1}^{k-1} \lambda_k}{\sum_{j=1}^{k} q_{\sigma(j)}} \; , \tag{3.21}$$

where

$$1 \quad \geq \quad \lambda_j \quad \stackrel{\text{def.}}{=} \quad \exp\left\{ \frac{\sum_{j=1}^{k} q_{\sigma(j)}}{T^{(2t-1)}} \left[\left(M_{i,\sigma(j)} + L_{\sigma(j)}^{(t)} \right) \; - \; \left(M_{i,\sigma(j+1)} + L_{\sigma(j)}^{(t+1)} \right) \right] \right\}$$

for $i = 1, \ldots, m_2 - 1$ *and* $\lambda_{m_2} = 0$. *Then, the conditional expectation*

$$\mathbb{E}\left[U_i^{(t)} \middle| \mathbf{L}^{(t)} \right] = M_{i,\sigma(1)} + L_{\sigma(1)}^{(t)} + \psi_i^{(t)} T^{(2t-1)} \; .$$

In particular, we denote $\sigma(1)$ *by* J_i^t *or* $J(i,t)$.

We note that the calculation of Eq. (3.20) and Eq. (3.21) needs $O(m_1 \log m_1)$ and $O(m_2 \log m_2)$ time respectively. By a few additional calculations, we introduce the notation $\mathcal{O}(\cdot, \cdot)$:

$$\begin{aligned}
\mathcal{O}(\mathbf{z}^{2t}, T^{(2t)}) \quad &\stackrel{\text{def.}}{=} \quad \mathbb{E}\left[\langle \mathbf{q}, \mathbf{L}^{(t)} \rangle - \langle \mathbf{q}, \mathbf{L}^{(t+1)} \rangle \middle| \mathbf{U}^{(t)}, \mathbf{L}^{(t)} \right] \\
&= \quad \sum_{j=1}^{m_2} \left(M_{I_j^t, j} + L_j^{(t)} - U_{I_j^t}^{(t)} + \phi_j^{(t)} T^{(2t)} \right) q_j \\
\mathcal{O}(\mathbf{z}^{2t-1}, T^{(2t-1)}) \quad &\stackrel{\text{def.}}{=} \quad \mathbb{E}\left[\langle \mathbf{p}, \mathbf{U}^{(t)} \rangle - \langle \mathbf{p}, \mathbf{U}^{(t-1)} \rangle \middle| \mathbf{U}^{(t-1)}, \mathbf{L}^{(t)} \right] \\
&= \quad \sum_{i=1}^{m_1} \left(M_{i, J_i^t} + L_{J_i^t}^{(t)} - U_i^{(t-1)} + \psi_i^{(t)} T^{(2t-1)} \right) p_i
\end{aligned} \tag{3.22}$$

Note that $\mathcal{O}(\mathbf{z}^n, T^n) = \mathbb{E}\left[V(\mathbf{z}^{n+1}) - V(\mathbf{z}^n)|\mathbf{z}^n\right]$.

Recovery of Approximate Primal Solution. An approximate (m_1+m_2)-sparse primal solution[3] can be recovered from $\mathbf{z}^n$ at $n = 2t$ by

$$Z \approx \frac{1}{2}\text{sparse}(1:m_1, J(1:m_1,t), \mathbf{p}) + \frac{1}{2}\text{sparse}(I(1:m_2,t), 1:m_2, \mathbf{q}) \in \mathbb{R}^{m_1\times m_2}. \quad (3.23)$$

Concentration Bounds. We are interested in the concentration bound related to $V(\mathbf{z}^n)$ because it replaces the true Wasserstein loss in WLMs. Given $\mathbf{U}^{(0)}$ (*i.e.*, $\mathbf{z}^1$ is implied), for $n = 1, \ldots, 2N$, we let

$$r^n = V(\mathbf{z}^n) - \sum_{s=1}^{n-1} \mathcal{O}(\mathbf{z}^s, T^{(s)}) \, . \quad (3.24)$$

This is crucial for one who wants to know whether the cooling schedule is too fast to secure the suboptimality within a finite budget of iterations. The following Theorem 3.4.2 gives a possible route to approximately realize this goal. It bounds the difference between

$$V(\mathbf{z}^n) - V(\mathbf{z}^1) \text{ and } \sum_{s=1}^{n-1} \mathbb{E}\left[V(\mathbf{z}^{s+1}) - V(\mathbf{z}^s)|\mathbf{z}^s\right],$$

the second of which is a quantitative term representing sum of a sequence. We see that $\mathcal{O}(\mathbf{z}^s, T^{(s)}) = \mathbb{E}\left[V(\mathbf{z}^{s+1}) - V(\mathbf{z}^s)|\mathbf{z}^s\right] = 0$ if and only if $T^{(s)} = \mathcal{T}(\mathbf{z}^s) \overset{\text{def.}}{=}$

$$\begin{cases} -\dfrac{1}{\langle \phi^{(t)}, \mathbf{q}\rangle} \sum\limits_{j=1}^{m_2} q_j \left[M_{I_j^t, j} + L_j^{(t)} - U_{I_j^t}^{(t)}\right] & \text{if } s = 2t \\ -\dfrac{1}{\langle \psi^{(t)}, \mathbf{p}\rangle} \sum\limits_{i=1}^{m_1} p_i \left[M_{i, J_i^t} + L_{J_i^t}^{(t)} - U_i^{(t-1)}\right] & \text{if } s = 2t-1 \end{cases} \quad (3.25)$$

In the practice of Gibbs-OT, choosing the proper cooling schedule for a specific WLM needs trial-and-error. Here we present a heuristics that the temperature $T^{(s)}$ is often chosen and adapted around $\eta\mathcal{T}(z^s)$, where $\eta \in [0.1, 0.9]$. We have two concerns regarding the choice of temperature T: First, in a WLM, the cost $V(\mathbf{z})$ is to be gradually minimized, hence a temperature T smaller than $\mathcal{T}(\mathbf{z}^s)$

[3]The notation of sparse$(\cdot,\cdot,\cdot)$ function is introduced under the syntax of MATLAB: http://www.mathworks.com/help/matlab/ref/sparse.html

at every iteration ensures that the cost is actually decreased by expectation, *i.e.*, $\mathbb{E}[V(\mathbf{z}^n) - V(\mathbf{z}^1)] < 0$; second, if T is too small, it takes many iterations to reach a highly accurate equilibrium, which might not be necessary for a single outer level step of parameter update.

Theorem 3.4.2 (Concentration bounds for finite time Gibbs-OT)**.** *First, r^n (by definition) is a martingale subject to the filtration of $\mathbf{z}_1, \ldots, \mathbf{z}_n$. Second, given a $\varepsilon \in (0,1)$, for $n = 1, \ldots, 2N-1$ if we choose the temperature schedule $T^{(1)}, \ldots, T^{(2N)}$ such that (i) $C^n \cdot T^{(n)} \leq a_n$, or (ii) $\exists \gamma > 0$, $\log\left(\frac{2N \max\{m_1, m_2\}}{\varepsilon}\right) \cdot T^{(n)} + D^n \leq \gamma a_n$, where $\{a_n \geq 0\}$ is a pre-determined array. Here for $t = 1, \ldots, N$,*

$$\begin{aligned} C^{2t-1} &\overset{\text{def.}}{=} \langle \psi^{(t)}, \mathbf{p} \rangle \,, \\ C^{2t} &\overset{\text{def.}}{=} \langle \phi^{(t)}, \mathbf{q} \rangle \,, \\ D^{2t-1} &\overset{\text{def.}}{=} \sum_{i=1}^{m_1} p_i \mathcal{R}\left(M_{i,\cdot}^T + \mathbf{L}^{(t)}; \mathbf{q}\right) \,, \\ D^{2t} &\overset{\text{def.}}{=} \sum_{i=1}^{m_2} q_j \mathcal{R}\left(M_{\cdot,j} - \mathbf{U}^{(t)}; \mathbf{p}\right) \,, \end{aligned}$$

where $M_{i,\cdot}$ and $M_{\cdot,j}$ represents the i-th rows and j-th columns of matrix M respectively, $\psi^{(t)}$ and $\phi^{(t)}$ are defined in Lemma 3.4.1, and regret function $\mathcal{R}(\mathbf{x}; \mathbf{w}) \overset{\text{def.}}{=} \sum_{i=1}^m w_i x_i - \min_{1 \leq i \leq m} x_i$ for any $\mathbf{w} \in \Delta_m$ and $\mathbf{x} \in \mathbb{R}^m$. Then for any $K > 0$, we have

$$Pr\left(r^{2N} < r^1 - K\right) \leq \exp\left[-\frac{K^2}{2\sum_{i=1}^{2N-1} a_n^2}\right] \,, \tag{3.26}$$

or

$$Pr\left(r^{2N} > r^1 + \gamma K\right) \leq \exp\left[-\frac{K^2}{2\sum_{i=1}^{2N-1} a_n^2}\right] + \varepsilon \,. \tag{3.27}$$

Remark 6. The bound obtained is a quantitative Hoeffding bound, not a bound that guarantees contraction around the true solution of dual OT. Nevertheless, we argue that this bound is still useful in investigating the proposed Gibbs sampler when the temperature is not annealed to zero. Particularly, the bound is for cooling schedules in general, *i.e.*, it is more applicable than a bound for a specific schedule. There has long been a gap between the practice and theory of SA despite of its wide usage. Our result likewise falls short of firm theoretical guarantee from the optimization perspective, as with the usual application of SA.

3.4.4 Proof of Lemmas and Theorem

The minimum of n independent exponential random variables with different parameters has computatable formula for its expectation. The result immediately lays out the proof of Lemma 3.4.1.

Lemma 3.4.3. *Suppose we have n independent exponential random variables e_i whose c.d.f. is by $f_i(x) = \min\{\exp(\omega_i(x - z_i)), 1\}$. Without lose of generality, we assume $z_1 \geq z_2 \geq ... \geq z_n$, then let $z_{n+1} = -\infty, h_i = \exp\left[\sum_{j=1}^{i} \omega_j(z_{i+1} - z_i)\right] \leq 1$ (with $h_n = 0, z_{n+1}h_n = 0$), we have*

$$\mathbb{E}\left[\max\{e_1, \ldots, e_n\}\right] = z_1 - \sum_{i=1}^{n} \frac{(1 - h_i)\prod_{j=1}^{i-1} h_i}{\sum_{j=1}^{i} \omega_j} \ .$$

Proof. The c.d.f. of $\max\{e_1, \ldots, e_n\}$ is $F(x) = \prod_{i=1}^{n} f_i(x)$ which is piece-wise smooth with interval (z_{i+1}, z_i), we want to calculate $\int_{-\infty}^{\infty} x dF(x)$.

$$\begin{aligned}
\int_{-\infty}^{\infty} x dF(x) &= \sum_{i=1}^{n} \int_{z_{i+1}}^{z_i} x dF(x) + 0 \\
&= \sum_{i=1}^{n} \int_{z_{i+1}}^{z_i} x d \exp\left[\sum_{j=1}^{i} \omega_j(x - z_j)\right] \\
&= \sum_{i=1}^{n} \int_{z_{i+1}}^{z_i} \left[\sum_{j=1}^{i} \omega_j\right] x \exp\left[\sum_{j=1}^{i} \omega_j(x - z_j)\right] dx \\
&= \sum_{i=1}^{n} \Bigg\{ \left(z_i - \frac{1}{\sum_{j=1}^{i} \omega_j}\right) \exp\left[\sum_{j=1}^{i} \omega_j(z_i - z_j)\right] \\
&\qquad - \left(z_{i+1} - \frac{1}{\sum_{j=1}^{i} \omega_j}\right) \exp\left[\sum_{j=1}^{i} \omega_j(z_{i+1} - z_j)\right] \Bigg\} \\
&= \sum_{i=1}^{n} \left[(z_i - z_{i+1}h_i) - \frac{1 - h_i}{\sum_{j=1}^{i} \omega_j}\right] \prod_{j=1}^{i-1} h_i \\
&= \sum_{i=1}^{n} \left[z_i \prod_{j=1}^{i-1} h_i - z_{i+1} \prod_{j=1}^{i} h_i\right] - \sum_{i=1}^{n} \frac{(1 - h_i)\prod_{j=1}^{i-1} h_i}{\sum_{j=1}^{i} \omega_j} \\
&= z_1 - \sum_{i=1}^{n} \frac{(1 - h_i)\prod_{j=1}^{i-1} h_i}{\sum_{j=1}^{i} \omega_j} \ .
\end{aligned}$$

□

Therefore Lemma 3.4.1 is proved up to trivial calculation using the above Lemma 3.4.3. In order to further prove Lemma 3.4.5, we also have (by definition of $F(x)$).

Lemma 3.4.4. *Subject to the setup of Lemma 3.4.3, we also have*

$$\max\{e_1, \ldots, e_n\} \leq z_1 \ ,$$

and

$$F(x) \leq \min\left\{\exp\left[\sum_{i=1}^{n} \omega_i (x - z^*)\right], 1\right\}, -\infty < x < \infty,$$

where $z^* = \dfrac{\sum_{i=1}^{n} \omega_i z_i}{\sum_{i=1}^{n} \omega_i}$.

Therefore, based on the observation of Lemma 3.4.4, the tail probability $Pr(\max\{e_1, \ldots, e_n\} < x)$ is upper bounded by the probability of an exponential random variable, which lead us to the proof of Lemma 3.4.5.

Lemma 3.4.5. *Note that Eq. (3.22) implies* $\mathbb{E}\left[r^{n+1} - r^n | \mathbf{z}^1, \ldots, \mathbf{z}^n\right] = 0$ *for* $t = 1, \ldots, 2N$. *Therefore,* $\{r^n\}$ *is a (discrete time) martingale subject to the filtration of* $\{\mathbf{z}^n\}$. *(Recall the notation by Eq. (3.19).) Moreover, we have the following two bounds. First, we can establish the left hand side bound for* $\{r^{n+1} - r^n\}_{n=1}^{2N-1}$*:*

$$r^n - r^{n+1} \leq C^n \cdot T^{(n)},$$

where for $t = 1, \ldots, N$

$$C^{2t-1} \stackrel{\text{def.}}{=} \langle \psi^{(t)}, \mathbf{p} \rangle \ \textit{and} \ C^{2t} \stackrel{\text{def.}}{=} \langle \phi^{(t)}, \mathbf{q} \rangle. \tag{3.28}$$

Second, we also bound on the right hand side. That said, for any $1 > \varepsilon > 0$, *we have*

$$Pr\Big(\exists n \in \{1, \ldots, 2N\}, \ s.t. \ r^{n+1} - r^n \geq \log\left(\frac{2N \max\{m_1, m_2\}}{\varepsilon}\right) \cdot T^{(n)} + D^n \Big| \mathbf{z}^1, \ldots, \mathbf{z}^n \Big) \leq \varepsilon, \tag{3.29}$$

where for $t = 1, \ldots, N$

$$D^{2t-1} \stackrel{\text{def.}}{=} \sum_{i=1}^{m_1} p_i \mathcal{R}\left(M_{i,\cdot}^T + \mathbf{L}^{(t)}; \mathbf{q}\right) \tag{3.30}$$

$$D^{2t} \stackrel{\text{def.}}{=} \sum_{i=1}^{m_2} q_j \mathcal{R}\left(M_{\cdot,j} - \mathbf{U}^{(t)}; \mathbf{p}\right), \tag{3.31}$$

where $M_{i,\cdot}$ *and* $M_{\cdot,j}$ *represents the* i*-th rows and* j*-th columns of matrix* M *respectively.*

Proof. On one hand, because for each $i \in \{1, \ldots, m_1\}$, $U_i^{(t)}|\mathbf{L}^{(t)}$ is lower bounded by $M_{i,J(i,t)} + L_{J(i,t)}^{(t)}$ (Lemma 3.4.4), and for each $j \in \{1, \ldots, m_2\}$, $L_j^{(t)}|\mathbf{U}^{(t-1)}$ is upper bounded by $U_{I(j,t)}^{(t-1)} - M_{I(j,t),j}$ (Lemma 3.4.4), we easily (by definition) have $r^{n+1}|\mathbf{z}_1, \ldots, \mathbf{z}^n$ is lower bounded by $r^n - C^n \cdot T^{(n)}$.

On the other hand, we have if $r^{n+1} - r^n \geq \log(1/\varepsilon_0) \cdot T^{(n)} + D^n|\mathbf{z}_1, \ldots, \mathbf{z}_n$ for some $\varepsilon_0 > 0$, then at least one of $U_i^{(t)}$ (or $L_j^{(t)}$) violates the bound $\log(1/\varepsilon_0) \cdot T^{(n)} + \mathcal{R}(M_{i,\cdot}^T + \mathbf{L}^{(t)}; \mathbf{q})$ (or $\log(1/\varepsilon_0) \cdot T^{(n)} + \mathcal{R}(M_{\cdot,j} - \mathbf{U}^{(t)}; \mathbf{p})$), whose probability using Lemma 3.4.4 is shown to be less than ε_0. Therefore, we have for each n

$$Pr(r^{n+1} - r^n \geq \log(1/\varepsilon_0) \cdot T^{(n)} + D^n|\mathbf{z}_1, \ldots, \mathbf{z}_n) \leq \max\{m_1, m_2\}\varepsilon_0 \ , \tag{3.32}$$

and

$$Pr(\exists n, r^{n+1} - r^n \geq \log(1/\varepsilon_0) \cdot T^{(n)} + D^n|\mathbf{z}_1, \ldots, \mathbf{z}_n) \leq 2N \max\{m_1, m_2\}\varepsilon_0 \ , \tag{3.33}$$

Let $\varepsilon = 2N \max\{m_1, m_2\}\varepsilon_0$, which concludes our result. □

Given Lemma 3.4.5, we can prove Theorem 3.4.2 by applying the classical Azuma's inequality for the left-hand side bound, and applying one of its extensions (Proposition 34 in (Tao and Vu, 2015)) for the right-hand side bound. Remark that Theorem 3.4.2 is about a single OT. For multiple different OTs, which share the same temperature schedule, one can have asymptotic bounds using the Law of Large Numbers due to the fact that their Gibbs samplers are independent with each other. Let $R^n = \frac{1}{S}\sum_{k=1}^{S} r_k^n$, where r_k^n is defined by Eq. (3.24) for sample k. Since for any $\varepsilon > 0$, one has $P(|R^{n+1} - R^n| > \varepsilon) \to 0$, as $S \to \infty$, one can have

the asymptotic concentration bound for R^{2N} that for any $\varepsilon_1, \varepsilon_2 > 0$, there exists S such that $P(\left|R^{2N} - R^1\right| > \varepsilon_1) \leq \exp\left(-\frac{1}{2N\varepsilon_2}\right)$.

3.5 Toy OT Examples

1D Case with Euclidean Cost. We first illustrate the differences between the approximate primal solutions computed by different methods by replicating a toy example in [21]. The toy example calculates the OT between two 1D two-mode distributions. We visualize their solved coupling as a 2D image in Fig. 3.2 at the budgets in terms of different number of iterations. Given their different convergence behaviors, when one wants to compromise with using pre-converged primal solutions in WLMs, he or she has to account for the different results computed by different numerical methods, even though they all aim at the Wasserstein loss.

As a note, Sinkhorn, B-ADMM and Gibbs-OT share the same computational complexity per iteration. The difference in their actual CPU time comes from the different arithmetic operations used. B-ADMM may be the slowest because it requires `log()` and `exp()` operations. When memory efficiency is of concern, both the implementations of Sinkhorn and Gibbs-OT can be modified to take only $O(m_1 + m_2)$ additional memory besides the space for caching the cost matrix M.

Two Electrons with Coulomb Cost in DFT. In quantum mechanics, Coulomb cost (or electron-electron Coulomb repulsion) is an important energy functional in Density Functional Theory (DFT). Numerical methods that solve the multi-marginal OT problem with unbounded costs remains an open challenge in DFT [36]. We consider two uniform densities on 1D domain [0, 1] with Coulomb cost $c(x, y) = 1/|x - y|$ which has analytic solutions. Coulumb cost is different from the usual metric cost in the OT literature, which is unbounded and singular at $x = y$. As observed in [36], the entropic regularized primal solution becomes more concentrated at boundaries, which is not physically plausible. This effect is not observed in the Gibbs-OT solution as shown in Fig. 3.3. As shown by Fig 3.1, the variables $\mathbf{U}, \mathbf{V}$ in computation are always in bounded range (with an overwhelming probability), thus the algorithm does not endure any numerical difficulties.

For entropic regularization [21, 36], we empirically select the minimal ε which does not cause numerical overflow before 5000 iterations (in which $\varepsilon = 0.5/N$). For Gibbs-OT, we use a geometric temperature scheme such that $T = 2.0(1/l^4)^{n/l}/N$

at the n-th iteration, where l is the max iteration number. For the unbounded Coulomb cost, Bregman ADMM [31] does not converge to a solution close to the true optimum.

Chapter 4
Unsupervised Wasserstein Learning

4.1 Overview

This chapter will be devoted to two unsupervised learning models for distributions. One is about clustering and the other is about component analysis. Like conventional methods for vectors such as K-means and non-negative matrix factorization, One can imagine there must exist counterparts in the space of Wasserstein learning.

4.2 Wasserstein Barycenter Problem and Discrete Distribution Clustering

Distribution clustering can be subjected to different affinity definitions. For example, Bregman clustering pursues the minimal distortion between the cluster prototype, called the Bregman representative, and cluster members according to a certain Bregman divergence [37]. In comparison, D2-clustering is an extension of K-means to discrete distributions under the Wasserstein distance [19], and the cluster prototype is an approximate Wasserstein barycenter with a sparse finite support set. In the D2-clustering framework, solving the cluster prototype or the centroid for discrete distributions under the Wasserstein distance is computationally challenging [2,11,38]. In order to scale up the computation of D2-clustering, a divide-and-conquer approach

has been proposed [38], but the method is ad-hoc from an optimization perspective. A standard ADMM approach has also been explored [11], but its efficiency is still inadequate for large datasets. Although fast computation of Wasserstein distances has been much explored [28, 31, 39], how to perform top-down clustering efficiently based on the distance has not.

The centroid of a collection of distributions minimizing the average pth-order power of the L_p Wasserstein distance is called Wasserstein barycenter [20]. In the D2-clustering algorithm [19], the 2nd-order Wasserstein barycenter is simply referred to as a prototype or centroid, and is solved for the case of an unknown support with a pre-given cardinality. The existence, uniqueness, regularity and other properties of the 2nd-order Wasserstein barycenter have been established mathematically for continuous measures in the Euclidean space [20]. The situation for discrete distributions, however, is more intricate, as will be explained later.

Given N arbitrary discrete distributions each with $\bar{m}$ support points, their true Wasserstein barycenter in theory can be solved via linear programming [20, 40]. This approach is logical because the support points of the Wasserstein barycenter can only locate at a finite (yet huge) number of possible positions. Yet, solving the true discrete barycenter quickly becomes intractable even for a rather small number of distributions containing only 10 support points each. Anderes *et al.* [40] made important theoretical progress on this particular challenge by proving that the actual support of a true barycenter of N such distributions is extremely sparse, with cardinality m no greater than $\bar{m}N$. Unfortunately, the complexity of the problem is not reduced practically because so far there is no theoretically ensured way to sift out the optimal sparse locations. Anderes *et al.*'s approach, though, does backup the practice of assuming a pre-selected number of support points in a barycenter as an approximation to the true solution.

To achieve good approximation, two computational strategies are useful in an optimization framework.

(i) Carefully select beforehand a large and representative set of support points as an approximation to the support of the true barycenter (*e.g.*, K-means).

(ii) Allow the support points in a barycenter to adjust positions at every τ iterations.

The first strategy of fixing the support of a barycenter can yield adequate approxi-

mation quality in low dimensions (*e.g.* 1D/2D histogram data) [2, 21], but can face the challenge of an exponentially growing support size when the dimension increases. The second strategy allows one to use a possibly much smaller number of support points in a barycenter to achieve the same level of accuracy [2,11,19,38]. Because the time complexity per iteration of existing iterative methods is $O(\bar{m}mN)$, a smaller m can also save much computation, and the extra amount of time $O(\bar{m}mdN/\tau)$ can be used to recalculate the distance matrices. In the extreme case when the barycenter support size is set to one ($m = 1$), D2-clustering reduces to K-means on the distribution means, which is a meaningful way of data reduction in its own right. Our experiments indicate that in practice a large m in D2-clustering is usually unnecessary (see Section 4.3.6 for related discussions).

In applications on high-dimensional data, optimizing the support points is preferred to fixing them from the beginning. This option, however, leads to a non-convex optimization problem. Our work aims at developing practical numerical methods. In particular, our method optimizes jointly the locations and weights of the support points in a single loop without resorting to a bi-level optimization reformulation, as was done in earlier work [2, 19].

4.2.1 Discrete Wasserstein Barycenter in Different Data Settings

Recently, a series of works have been devoted to solving the Wasserstein barycenter given a set of distributions (*e.g.* [2, 3, 11, 21, 41]). How our method compares with the existing ones depends strongly on the specific data setting. We discuss the comparisons in details below and promote the use of our new method, AD2-clustering.

In [2, 21, 28], novel algorithms have been developed for solving the Wasserstein barycenter by adding an entropy regularization term on the optimal transport matching weights. The regularization is imposed on the transport weights, but not the barycenter distribution. In particular, iterative Bregman projection (IBP) [21] can solve an approximation to the Wasserstein barycenter. IBP is highly memory efficient for distributions with a shared support set (*e.g.* histogram data), with a memory complexity $O((m + \bar{m})N)$. In comparison, our modified B-ADMM

approach is of the same time complexity, but requires $O(m\bar{m}N)$ memory. If N is large, memory constraints can limit our approach to problems with relatively small m or $\bar{m}$. While the first strategy may not meet the memory constraint, the second approximation strategy is crucial for reaching high accuracy with our approach. Conventional OT literature emphasizes computing the Wasserstein barycenter for a small number of instances with dense representations (*e.g.* [42, 43]); and IBP is more suitable. Yet in many machine learning and signal processing applications, each instance is represented by a discrete distribution with a sparse finite support set (*i.e.*, $\bar{m}$ is small). The memory limitation of B-ADMM can be avoided via parallelization until the time allocation is spent. Our focus is thus to achieve scalability in N.

As demonstrated by experiments, B-ADMM has advantages over IBP that motivate its usage in our algorithm. If the distributions do not share the support set, IBP has the same memory complexity $O(m\bar{m}N)$ (for caching the distance matrix per instance) as our approach has. In addition, B-ADMM [31], based on which our approach is developed, has the following advantages: (1) It yields the exact OT and distance in the limit of iterations. Note that the ADMM parameter does not trade off the convergence rate. (2) It requires little tuning of hyper-parameters and easily accommodates warm starts (to be illustrated later), which are valuable traits for the task of D2-clustering. (3) It works well with single-precision floats, and thus it is not restricted by the machine precision constraint. In contrast, IBP requires more tuning and may encounter precision overflow which is hard to address. Our experiments show that when its coupling solutions are used, the resulting discrete Wasserstein barycenters with sparse finite support sets are not as accurate as those by B-ADMM (see [21] and our experiments).[1]

Our main algorithm is inspired by the B-ADMM algorithm of Wang and Banerjee [31] for solving OT fast. They developed the two-block variant of ADMM [44] along with Bregman divergence to solve OT when the number of support points is extremely large. Its algorithmic relation to IBP [21] is discussed in Section 4.4. The OT problem at a moderate scale can in fact be efficiently handled by state-of-the-art LP solvers [45]. As demonstrated by the line of work on solving the barycenter, optimizing the Wasserstein barycenter is rather different from computing the distance. Whereas naïvely adapting the B-ADMM to Wasserstein barycenter does not

[1] Here "accurate" means close to a local minimizer of sum of the (squared) Wasserstein distances.

result in a proper algorithm, our modified B-ADMM algorithm effectively addresses the Wasserstein barycenter problem. The modification we made on B-ADMM is necessary. Although the modified B-ADMM approach is not guaranteed to converge to a local optimum, it often yields a solution very close to the local optimum. The new method is shown empirically to achieve higher accuracy than IBP or its derivatives when distributions are supported on finite sets.

Finally, we note that although solving a single barycenter for a fixed set is a key component in D2-clustering, the task of clustering per se bears some extra technical subtleties. In a clustering setup, the partition of samples varies over the iterations, and a sequence of Wasserstein barycenters are solved. We found that the robustness with respect to the hyper-parameters in the optimization algorithms is as important as the speed of solving one centroid because tuning these parameters over many iterations of different partitions is impractical.

4.2.2 D2-Clustering

Consider discrete distributions with sparse finite support specified by a set of support points and their associated probabilities, a.k.a. weights:

$$\{(w_1, x_1), \ldots, (w_m, x_m)\},$$

where $\sum_{i=1}^{m} w_i = 1$ with $w_i \geqslant 0$, and $x_i \in \mathbb{M}$ for $i = 1, \ldots, m$. Usually, $\mathbb{M} = \mathbb{R}^d$ is the d-dimensional Euclidean space with the L_p norm, and x_i's are also called support vectors. $\mathbb{M}$ can also be a symbolic set provided with symbol-to-symbol dissimilarity. The Wasserstein distance between distributions $P^{(a)} = \{(w_i^{(a)}, x_i^{(a)}), i = 1, ..., m_a\}$ and $P^{(b)} = \{(w_i^{(b)}, x_i^{(b)}), i = 1, ..., m_b\}$ is solved by the following linear programming (LP). For notation brevity, let $c(x_i^{(a)}, x_j^{(b)}) = \|x_i^{(a)} - x_j^{(b)}\|_p^p$. Define index set $\mathcal{I}_a = \{1, ..., m_a\}$ and $\mathcal{I}_b$ likewise. We define $\left(W_p(P^{(a)}, P^{(b)})\right)^p :=$

$$\begin{aligned} \min_{\{\pi_{i,j} \geqslant 0\}} \quad & \sum_{i \in \mathcal{I}_a, j \in \mathcal{I}_b} \pi_{i,j} c(x_i^{(a)}, x_j^{(b)}) \ , \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{m_a} \pi_{i,j} = w_j^{(b)}, \ \forall j \in \mathcal{I}_b \ , \\ & \textstyle\sum_{j=1}^{m_b} \pi_{i,j} = w_i^{(a)}, \ \forall i \in \mathcal{I}_a \ . \end{aligned} \tag{4.1}$$

We call $\{\pi_{i,j}\}$ the *matching weights* between support points $x_i^{(a)}$ and $x_j^{(b)}$ or the

optimal coupling for $P^{(a)}$ and $P^{(b)}$. In D2-clustering, we use the L_2 Wasserstein distance. From now on, we will denote W_2 simply by W.

Consider a set of discrete distributions $\{P^{(k)}, k = 1, ..., \bar{N}\}$, where $P^{(k)} = \{(w_i^{(k)}, x_i^{(k)}), i = 1, ..., m_k\}$. The goal of D2-clustering is to find a set of centroid distributions $\{Q^{(i)}, i = 1, ..., K\}$ such that the total within-cluster variation is minimized:

$$\min_{\{Q^{(i)}\}} \sum_{k=1}^{\bar{N}} \min_{i=1,...,K} W^2(Q^{(i)}, P^{(k)}) \ .$$

Similarly as in K-means, D2-clustering alternates the optimization of the centroids $\{Q^{(i)}\}$ and the assignment of each instance to the nearest centroid, the iteration referred to as the *outer loop* (Algorithm 4.1). The major computational challenge in the algorithm is to compute the optimal centroid for each cluster at each iteration. This computation also marks the main difference between D2-clustering and K-means in which the optimal centroid is in a simple closed form. The new scalable algorithms we develop here aim primarily to expedite this optimization step. For clarity of presentation, we now focus on this optimization problem and describe the notation below. Suppose we have a set of discrete distributions $\{P^{(1)}, \ldots, P^{(N)}\}$. N is the sample size for computing one Wasserstein barycenter. We want to find a centroid $P : \{(w_1, x_1), \ldots, (w_m, x_m)\}$, such that

$$\min_{P} \frac{1}{N} \sum_{k=1}^{N} W^2(P, P^{(k)}) \tag{4.2}$$

with respect to the weights and support points of P. This is the **main question** we tackle in this paper. There is an implicit layer of optimization in (4.2)—the computation of $W^2(P, P^{(k)})$. The variables in optimization (4.2) thus include the weights in the centroid $\{w_i \in \mathbb{R}^+\}$, the support points $\{x_i \in \mathbb{R}^d\}$, and the optimal coupling between P and $P^{(k)}$ for each k, denoted by $\{\pi_{i,j}^{(k)}\}$ (see Eq. (4.1)).

To solve (4.2), D2-clustering alternates the optimization of $\{w_i\}$ and $\{\pi_{i,j}^{(k)}\}$, $k = 1, ..., N$, versus $\{x_i\}$.

1. Δ_k denotes a probability simplex of k dimensions.

2. $\mathbf{1}$ denotes a vector with all elements equal to one.

3. $\boldsymbol{x} = (x_1, \ldots, x_m) \in \mathbb{R}_{d\times m}$, $\boldsymbol{w} = (w_1, \ldots, w_m) \in \Delta_m$.

4. $\boldsymbol{x}^{(k)} = (x_1^{(k)}, \ldots, x_{m_k}^{(k)}) \in \mathbb{R}_{d\times m_k}$, $k = 1, ..., N$.

5. $\boldsymbol{w}^{(k)} = (w_1^{(k)}, \ldots, w_{m_k}^{(k)}) \in \Delta_{m_k}$.

6. $C(\boldsymbol{x}, \boldsymbol{x}^{(k)}) = (\|x_i - x_j^{(k)}\|^2)_{i,j} \in \mathbb{R}_{m\times m_k}$.

7. $\boldsymbol{X} = (\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}) \in \mathbb{R}_{d\times n}$, where $n = \sum_{k=1}^{N} m_k$.

8. $\Pi^{(k)} = (\pi_{i,j}^{(k)}) \in \mathbb{R}^+_{m\times m_k}$, $k = 1, ..., N$.

9. $\Pi = (\Pi^{(1)}, \ldots, \Pi^{(N)}) \in \mathbb{R}^+_{m\times n}$.

10. Index set $\mathcal{I}^c = \{1, ..., N\}$, $\mathcal{I}_k = \{1, ..., m_k\}$, for $k \in \mathcal{I}^c$, and $\mathcal{I}' = \{1, ..., m\}$.

With $\boldsymbol{w}$ and Π fixed, the cost function (4.2) is quadratic in terms of $\boldsymbol{x}$, and the optimal $\boldsymbol{x}$ is solved by:

$$x_i := \frac{1}{Nw_i} \sum_{k=1}^{N} \sum_{j=1}^{m_k} \pi_{i,j}^{(k)} x_j^{(k)}, \quad i \in \mathcal{I}' , \tag{4.3}$$

Or, we can write it in matrix form: $\boldsymbol{x} := \frac{1}{N} X\Pi^T \operatorname{diag}(1./\boldsymbol{w})$. However, with fixed $\boldsymbol{x}$, updating $\boldsymbol{w}$ and Π is challenging. D2-clustering solves a large LP as follows:

$$\begin{aligned} \min_{\Pi\in\mathbb{R}^+_{m\times n}, \boldsymbol{w}\in\Delta_m} & \sum_{k=1}^{N} \langle C(\boldsymbol{x}, \boldsymbol{x}^{(k)}), \Pi^{(k)} \rangle , \\ \text{s.t.} \quad & \mathbf{1} \cdot (\Pi^{(k)})^T = \boldsymbol{w} , \quad \mathbf{1} \cdot \Pi^{(k)} = \boldsymbol{w}^{(k)}, \forall k \in \mathcal{I}^c , \end{aligned} \tag{4.4}$$

where the inner product $\langle A, B\rangle := \operatorname{tr}(AB^t)$.

By iteratively solving (4.3) and (4.4), referred to as the *inner loop*, the step of updating the cluster centroid in Algorithm 4.1 is fulfilled [2, 19]. We present the centroid update step in Algorithm 4.2. In summary, D2-clustering is given by Algorithm 4.1 with Algorithm 4.2 embedded as one key step.

The major difficulty in solving (4.4) is that a standard LP solver typically has a polynomial complexity in terms of the number of variables $m + \sum_{k=1}^{N} m_k m$, prohibiting its scalability to a large number of discrete distributions in one cluster. When the cluster size is small or moderate, say dozens, it is shown that the standard LP solver can be faster than a scalable algorithm [11]. However, when

Algorithm 4.1 D2 Clustering [19]

procedure D2CLUSTERING($\{P^{(k)}\}_{k=1}^{M}$, K)
  Denote the label of each objects by $l^{(k)}$.
  Initialize K random centroid $\{Q^{(i)}\}_{i=1}^{K}$.
  repeat
    for $k = 1, \ldots, M$ **do** ▷ Assignment Step
      $l^{(k)} := \operatorname{argmin}_i W(Q^{(i)}, P^{(k)})$;
    end for
    for $i = 1, \ldots, K$ **do** ▷ Update Step
      $Q^{(i)} := \operatorname{argmin}_Q \sum_{l^{(k)}=i} W(Q, P^{(k)})$ (*)
    end for
  until the number of changes of $\{l^{(k)}\}$ meets some stopping criterion
  return $\{l^{(k)}\}_{k=1}^{M}$ and $\{Q^{(i)}\}_{i=1}^{K}$.
end procedure

Algorithm 4.2 Centroid Update with Full-batch LP [2, 19]

procedure CENTROID($\{P^{(k)}\}_{k=1}^{N}$)
  repeat
    Updates $\{x_i\}$ from Eq. (4.3);
    Updates $\{w_i\}$ from solving full-batch LP (4.4);
  until P converges
  return P
end procedure

the cluster size grows, the standard solver slows down quickly. This issue has been demonstrated by multiple empirical studies [11, 19, 38].

Our key observation is that in the update of a centroid distribution, the variables in $\boldsymbol{w}$ are much more important than are the matching weights in Π needed for computing the Wasserstein distances. The parameter $\boldsymbol{w}$ is actually part of the output centroid, while Π is not, albeit accounting for the vast majority of the variables in (4.4). We also note that the solution (4.4) is not the end result, but rather it is one round of centroid update in the outer loop. It is thus adequate to have a sufficiently accurate solution to (4.4), motivating us to pursue scalable methods such as ADMM, known to be fast for reaching the vicinity of the optimal solution.

4.3 Scalable Centroid Computation for D2-Clustering

We propose algorithms scalable with large-scale datasets, and compare their performance in terms of speed and memory. They are (a) subgradient descent with N mini-LP following similar ideas of [2] (included in Appendix 4.3.1), (b) standard ADMM with N mini-QP, and (c) modified B-ADMM with closed forms in each iteration of the inner loop. The bottleneck in the computation of D2-clustering is the inner loop, detailed in Algorithm 4.2. The approaches we develop here all aim for fast solutions for the inner loop, that is, to improve Algorithm 4.2. These new methods can reduce the computation for centroid update to a comparable (or even lower) level as the label assignment step, usually negligible in the original D2-clustering. As a result, we also take measures to expedite the labeling step, with details provided in Section 4.3.4.

4.3.1 Subgradient Descent Method

We describe a subgradient descent approach in this section for the purpose of experimental comparison.

Eq. (4.4) can be casted as multi-level optimization by treating $\boldsymbol{w}$ as policies/-parameters and Π as variables. Express $W^2(P, P^{(k)})$, the squared distance between P and $P^{(k)}$, as a function of $\boldsymbol{w}$ denoted by $\tilde{W}(\boldsymbol{w})^{(k)}$. $\tilde{W}(\boldsymbol{w})^{(k)}$ is the solution to a designed optimization, but has no closed form. Let $\tilde{W}(\boldsymbol{w}) = \frac{1}{N}\sum_{k=1}^{N} \tilde{W}(\boldsymbol{w})^{(k)}$, where N is the number of instances in the cluster. Note that Eq. (4.4) minimizes $\tilde{W}(\boldsymbol{w})$ up to a constant multiplier. The minimization of $\tilde{W}$ with respect to $\boldsymbol{w}$ is thus a bi-level optimization problem. In the special case when the designed problem is LP and the parameters only appear on the right hand side (RHS) of the constraints or are linear in the objective, the subgradient, specifically $\nabla\tilde{W}(\boldsymbol{w})^{(k)}$ in our problem, can be solved via the same (dual) LP.

Again, we consider the routine that alternates the updates of $\{x_i\}$ and $\{\pi_{i,j}\}^{(k)}$ iteratively. With fixed $\{x_i\}$, updating $\{\pi_{i,j}\}^{(k)}$ involves solving N LP (4.1). With LP (4.1) solved, we can write $\nabla\tilde{W}(\boldsymbol{w})^{(k)}$ in closed form, which is given by the set

The work presented in this section has been published in the form of a research paper: Jianbo Ye, Panruo Wu, James Z. Wang and Jia Li, "Fast Discrete Distribution Clustering Using Wasserstein Barcenter with Sparse Support," *IEEE Transactions on Signal Processing (TSP)* , Vol 65, Issue 9, pp 2317–2332, 2017.

of dual variables $\{\lambda_i^{(k)}\}_{i=1}^m$ corresponding to $\left\{\sum_{j=1}^{m_k} \pi_{i,j} = w_i, i = 1, ..., m\right\}$. Because $\{w_i\}$ must reside in the facets defined by Δ_m, the projected subgradient $\nabla \tilde{W}(\boldsymbol{w})^{(k)}$ is given by

$$\nabla \tilde{W}(\boldsymbol{w})^{(k)} = (\lambda_1^{(k)}, \ldots, \lambda_m^{(k)}) - \left(\sum_{i=1}^{m} \lambda_i^{(k)}\right)(1, \ldots, 1) \,. \tag{4.5}$$

In the standard method of gradient descent, a line search is conducted in each iteration to determine the step-size for a strictly descending update. Line search however is computationally intensive for our problem because Eq. (4.5) requires solving a LP and we need Eq. (4.5) sweeping over $k = 1, ..., N$. In machine learning algorithms, one practice to avoid expensive line search is by using a pre-determined step-size, which is allowed to vary across iterations. We adopt this approach here.

One issue resulting from a pre-determined step-size is that the updated weight vector $\boldsymbol{w}$ may have negative components. We overcome this numerical instability by the technique of re-parametrization. Let

$$w_i(\boldsymbol{s}) := \frac{\exp(s_i)}{\sum \exp(s_i)} \,, \; i = 1, ..., m \,. \tag{4.6}$$

We then compute the partial subgradient with respect to s_i instead of w_i, and update w_i by updating s_i. Furthermore $\exp(s_i)$ are re-scaled in each iteration such that $\sum_{i=1}^m s_i = 0$.

The step-size $\alpha(\boldsymbol{w})$ is chosen by

$$\sigma(\boldsymbol{w}) := \min\left(\frac{\alpha}{\|\nabla_{\boldsymbol{s}} \tilde{W}(\boldsymbol{w}(\boldsymbol{s}))\|}, \zeta\right) \,. \tag{4.7}$$

The two hyper-parameters α and ζ trade off the convergence speed and the guaranteed decrease of the objective. Another hyper-parameter is τ which indicates the ratio between the update frequency of weights $\{w_i\}$ and that of support points $\{x_i\}$. In our experiments, we alternate one round of update for both $\{w_i\}$ and $\{x_i\}$. We summarize the subgradient descent approach in Algorithm 4.3. If the support points $\{x_i\}$ are fixed, the centroid optimization is a linear programming in terms of $\{w_i\}$. The subgradient descent method converges under mild conditions on the smoothness of the solution and small or adaptive step-sizes. In Algorithm 4.3, the support points are also updated, and the problem becomes non-convex.

Algorithm 4.3 Centroid Update with Subgradient Descent

procedure CENTROID($\{P^{(k)}\}_{k=1}^{N}$, P) ▷ with initial guess
  repeat
    Updates $\{x_i\}$ from Eq.(4.3) Every τ iterations;
    for $k = 1, \ldots, N$ **do**
      Obtain $\Pi^{(k)}$ and $\Lambda^{(k)}$ from LP: $W(P, P^{(k)})$
    end for
    $\nabla\tilde{W}(\boldsymbol{w}) := \frac{1}{N}\sum_{k=1}^{N} \nabla\tilde{W}(\boldsymbol{w})^{(k)}$; ▷ See Eq.(4.5)
    $s_i := s_i - \sigma(\boldsymbol{w})\nabla\tilde{W}(\boldsymbol{w}) \cdot \dfrac{\partial \boldsymbol{w}}{\partial s_i}$ ▷ See Eq. (4.7)
    $s_i := s_i - \sum_{j=1}^{m} s_j, i = 1, \ldots m$; ▷ rescaling step
    $w_i := \dfrac{\exp(s_i)}{\sum \exp(s_i)}, i = 1, \ldots m$; ▷ sum-to-one
  until P converges
  return P
end procedure

4.3.2 Alternating Direction Method of Multipliers

ADMM typically solves a problem with two sets of variables (in our case, they are Π and $\boldsymbol{w}$), which are only coupled in constraints, while the objective function is separable across this splitting of the two sets (in our case, $\boldsymbol{w}$ is not present in the objective function) [44]. Because problem (4.4) has multiple sets of constraints including both equalities and inequalities, it is not a typical scenario to apply ADMM. We propose to relax all equality constraints $\sum_{l=1}^{m_k} \pi_{i,l}^{(k)} = w_i$, $\forall k \in \mathcal{I}^c$, $i \in \mathcal{I}'$ in (4.4) to their corresponding augmented Lagrangians and use the other constraints to determine a convex set for the parameters being optimized. Let $\Lambda = (\lambda_{i,k})$, $i \in \mathcal{I}'$, $k \in \mathcal{I}^c$. Let ρ be a parameter to balance the objective function and the augmented Lagrangians. Define $\Delta_\Pi = \left\{(\pi_{i,j}^{(k)}) : \sum_{i=1}^{m} \pi_{i,j}^{(k)} = w_j^{(k)}, \pi_{i,j}^{(k)} \geqslant 0, k \in \mathcal{I}^c, i \in \mathcal{I}', j \in \mathcal{I}_k\right\}$. Recall that $\Delta_m = \{(w_1, \ldots, w_m) | \sum_{i=1}^{m} w_i = 1, w_i \geqslant 0\}$. As in the method of multipliers, we form the scaled augmented Lagrangian $L\rho(\Pi, \boldsymbol{w}, \Lambda)$ as follows

$$L_\rho(\Pi, \boldsymbol{w}, \Lambda) =$$
$$\sum_{k=1}^{N} \langle C(\boldsymbol{x}, \boldsymbol{x}^{(k)}), \Pi^{(k)} \rangle + \rho \sum_{\substack{i \in \mathcal{I}' \\ k \in \mathcal{I}^c}} \lambda_{i,k} \left(\sum_{j=1}^{m_k} \pi_{i,j}^{(k)} - w_i \right) + \frac{\rho}{2} \sum_{\substack{i \in \mathcal{I}' \\ k \in \mathcal{I}^c}} \left(\sum_{j=1}^{m_k} \pi_{i,j}^{(k)} - w_i \right)^2 . \quad (4.8)$$

Problem (4.4) can be solved using ADMM iteratively as follows.

$$\Pi^{n+1} := \underset{\Pi \in \Delta_\Pi}{\operatorname{argmin}} L_\rho(\Pi, \boldsymbol{w}^n, \Lambda^n) \ , \tag{4.9}$$

$$\boldsymbol{w}^{n+1} := \underset{\boldsymbol{w} \in \Delta_m}{\operatorname{argmin}} L_\rho(\Pi^{n+1}, \boldsymbol{w}, \Lambda^n) \ , \tag{4.10}$$

$$\lambda_{i,k}^{n+1} := \lambda_{i,k}^n + \sum_{j=1}^{m_k} \pi_{i,j}^{(k),n+1} - w_i^{n+1} \ , i \in \mathcal{I}', \ k \in \mathcal{I}^c \ . \tag{4.11}$$

Based on (4.9), Π can be updated by updating $\Pi^{(k)}$, $k = 1, ..., N$ separately. Comparing with the full batch LP in (4.4) which solves all $\Pi^{(k)}$, $k = 1, ..., N$, together, ADMM solves instead N disjoint constrained quadratic programming (QP). This step is the key for achieving computational complexity linear in N, the main motivation for employing ADMM. Specifically, we solve (4.4) by solving (4.12) below for each $k = 1, ..., N$:

$$\begin{aligned} \min_{\pi_{i,j}^{(k)} \geqslant 0} \quad & \langle C(\boldsymbol{x}, \boldsymbol{x}^{(k)}), \Pi^{(k)} \rangle + \frac{\rho}{2} \textstyle\sum_{i=1}^{m} \left(\sum_{j=1}^{m_k} \pi_{i,j}^{(k)} - w_i^n + \lambda_{i,k}^n \right)^2 \\ \text{s.t.} \quad & \mathbf{1} \cdot \Pi^{(k)} = \boldsymbol{w}^{(k)}, k \in \mathcal{I}^c. \end{aligned} \tag{4.12}$$

Since we need to solve small-size problem (4.12) in multiple rounds, we prefer active set method with warm start. Define $\tilde{w}_i^{(k),n+1} := \sum_{j=1}^{m_k} \pi_{i,j}^{(k),n+1} + \lambda_{i,k}^n$ for $i = 1, ..., m$, $k = 1, ..., N$. We can rewrite step (4.10) as

$$\min_{\boldsymbol{w} \in \Delta_m} \sum_{i=1}^{m} \sum_{k=1}^{N} (\tilde{w}_i^{(k),n+1} - w_i)^2 \ . \tag{4.13}$$

We summarize the computation of the centroid distribution P for distributions $P^{(k)}$, $k = 1, ..., N$, in Algorithm 4.4. There are two hyper-parameters to choose: ρ and the number of iterations T_{admm}. We empirically select ρ proportional to the averaged transportation costs:

$$\rho = \frac{\rho_0}{Nnm} \sum_{k=1}^{N} \sum_{i \in \mathcal{I}'} \sum_{j \in \mathcal{I}_k} c(x_i, x_j^{(k)}) \ . \tag{4.14}$$

Let us compare the computational efficiency of ADMM and the subgradient descent method. In gradient descent based approaches, it is costly to choose an

Algorithm 4.4 Centroid Update with ADMM [11]

procedure CENTROID($\{P^{(k)}\}_{k=1}^{N}$, P, Π)
  Initialize $\Lambda^0 = 0$ and $\Pi^0 := \Pi$.
  repeat
    Updates $\{x_i\}$ from Eq.(4.3);
    Reset dual coordinates Λ to zero;
    for $iter = 1, \ldots, T_{admm}$ **do**
      for $k = 1, \ldots, N$ **do**
        Update $\{\pi_{i,j}\}^{(k)}$ based on QP Eq.(4.12);
      end for
      Update $\{w_i\}$ based on QP Eq.(4.13);
      Update Λ based on Eq. (4.11);
    end for
  until P converges
  return P
end procedure

effective step-size along the descending direction because at each search point, we need to solve N LP — an issue also discussed in [21]. ADMM solves N QP sub-problems instead of LP. The amount of computation in each sub-problem of ADMM is thus usually higher and grows faster with the number of support points in $P^{(k)}$'s. Whether the increased complexity at each iteration of ADMM is paid off by a better convergence rate (*i.e.*, a smaller number of iterations) is unclear. The computational limitation of ADMM caused by QP motivates us to explore B-ADMM that avoids QP in each iteration.

4.3.3 Bregman ADMM

Bregman ADMM (B-ADMM) replaces the quadratic augmented Lagrangians by the Bregman divergence when updating the split variables [46]. Similar ideas trace back at least to the early 1990s [47, 48]. We adapt the design in [28, 31] for solving the OT problem with a large set of support points. Consider two sets of variables $\Pi_{(k,1)} = (\pi_{i,j}^{(k,1)})$, $i \in \mathcal{I}'$, $j \in \mathcal{I}_k$, and $\Pi_{(k,2)} = (\pi_{i,j}^{(k,2)})$, $i \in \mathcal{I}'$, $j \in \mathcal{I}_k$, for $k = 1, ..., N$ under the following constraints. Let

$$\Delta_{k,1} := \left\{ \pi_{i,j}^{(k,1)} \geqslant 0 : \sum_{i=1}^{m} \pi_{i,j}^{(k,1)} = w_j^{(k)}, j \in \mathcal{I}_k \right\} , \tag{4.15}$$

$$\Delta_{k,2}(\boldsymbol{w}) := \left\{ \pi_{i,j}^{(k,2)} \geqslant 0 : \sum_{j=1}^{m_k} \pi_{i,j}^{(k,2)} = w_i, i \in \mathcal{I}' \right\}, \tag{4.16}$$

then $\Pi^{(k,1)} \in \Delta_{k,1}$ and $\Pi^{(k,2)} \in \Delta_{k,2}(\boldsymbol{w})$. We introduce some extra notations:

1. $\bar{\Pi}^{(1)} = \{\Pi^{(1,1)}, \Pi^{(2,1)}, \ldots, \Pi^{(N,1)}\}$,
2. $\bar{\Pi}^{(2)} = \{\Pi^{(1,2)}, \Pi^{(2,2)}, \ldots, \Pi^{(N,2)}\}$,
3. $\bar{\Pi} = \{\bar{\Pi}^{(1)}, \bar{\Pi}^{(2)}\}$,
4. $\Lambda = \{\Lambda^{(1)}, \ldots, \Lambda^{(N)}\}$, where $\Lambda^{(k)} = (\lambda_{i,j}^{(k)})$, $i \in \mathcal{I}'$, $j \in \mathcal{I}_k$, is a $m \times m_k$ matrix.

B-ADMM solves (4.4) by treating the augmented Lagrangians conceptually as a designed divergence between $\Pi^{(k,1)}$ and $\Pi^{(k,2)}$, adapting to the updated variables. It restructures the original problem (4.4) as

$$\begin{aligned} \min_{\bar{\Pi}, \boldsymbol{w}} \quad & \sum_{k=1}^{N} \langle C(\boldsymbol{x}, \boldsymbol{x}^{(k)}), \Pi^{(k,1)} \rangle \ , \\ \text{s.t.} \quad & \boldsymbol{w} \in \Delta_m \ , \\ & \Pi^{(k,1)} \in \Delta_{k,1}, \quad \Pi^{(k,2)} \in \Delta_{k,2}(\boldsymbol{w}), \quad k = 1, \ldots, N \ , \\ & \Pi^{(k,1)} = \Pi^{(k,2)}, \quad k = 1, \ldots, N \ . \end{aligned} \tag{4.17}$$

Denote the dual variables $\Lambda^{(k)} = (\lambda_{i,j}^{(k)})$, $i \in \mathcal{I}'$, $j \in \mathcal{I}_k$, for $k = 1, ..., N$. Use $\mathrm{KL}(\cdot, \cdot)$ to denote the Kullback–Leibler divergence between two distributions. The B-ADMM algorithm adds the augmented Lagrangians for the last set of constraints in its updates, yielding the following equations.

$$\bar{\Pi}^{(1),n+1} := \underset{\{\Pi^{(k,1)} \in \Delta_{k,1}\}}{\operatorname{argmin}} \sum_{k=1}^{N} \left[\langle C(\boldsymbol{x}, \boldsymbol{x}^{(k)}), \Pi^{(k,1)} \rangle + \langle \Lambda^{(k),n}, \Pi^{(k,1)} \rangle + \rho \, \mathrm{KL}(\Pi^{(k,1)}, \Pi^{(k,2),n}) \right], \tag{4.18}$$

$$\bar{\Pi}^{(2),n+1}, \boldsymbol{w}^{n+1} := \underset{\substack{\{\Pi^{(k,2)} \in \Delta_{k,1}(\boldsymbol{w})\} \\ \boldsymbol{w} \in \Delta_m}}{\operatorname{argmin}} \sum_{k=1}^{N} \left[- \langle \Lambda^{(k),n}, \Pi^{(k,2)} \rangle + \rho \, \mathrm{KL}(\Pi^{(k,2)}, \Pi^{(k,1),n+1}) \right] \tag{4.19}$$

$$\Lambda^{n+1} := \Lambda^n + \rho(\bar{\Pi}^{(1),n+1} - \bar{\Pi}^{(2),n+1}). \tag{4.20}$$

We note that if $\boldsymbol{w}$ is fixed, (4.18) and (4.19) can be split by index $k = 1, ..., N$, and have closed form solutions for each k. Let *eps* be the floating-point tolerance (*e.g.*

10^{-16}). For any $i \in \mathcal{I}'$, $j \in \mathcal{I}_k$,

$$\tilde{\pi}_{i,j}^{(k,2),n} := \pi_{i,j}^{(k,2),n} \exp\left[\frac{c\left(x_i, x_j^{(k)}\right) + \lambda_{i,j}^{(k),n}}{-\rho}\right] + eps \ , \tag{4.21}$$

$$\pi_{i,j}^{(k,1),n+1} := \frac{\tilde{\pi}_{i,j}^{(k,2),n}}{\sum_{l=1}^{m} \tilde{\pi}_{l,j}^{(k,2),n}} \cdot w_j^{(k)} \ , \tag{4.22}$$

$$\tilde{\pi}_{i,j}^{(k,1),n+1} := \pi_{i,j}^{(k,1),n+1} \exp\left[\frac{1}{\rho}\lambda_{i,j}^{(k),n}\right] + eps \ , \tag{4.23}$$

$$\pi_{i,j}^{(k,2),n+1} := \frac{\tilde{\pi}_{i,j}^{(k,1),n+1}}{\sum_{l=1}^{m_k} \tilde{\pi}_{i,l}^{(k,1),n+1}} \cdot w_i \ . \tag{4.24}$$

Because we need to update $\boldsymbol{w}$ in each iteration, it is not easy to solve (4.19). We consider decomposing (4.19) into two stages. Observe that the minimum value of (4.19) under a given $\boldsymbol{w}$ is

$$\min_{\boldsymbol{w} \in \Delta_m} \sum_{k=1}^{N} \sum_{i=1}^{m} w_i \left[\log(w_i) - \log\left(\sum\nolimits_{j=1}^{m_k} \tilde{\pi}_{i,j}^{(k,1),n+1}\right)\right]. \tag{4.25}$$

The above term (a.k.a. the consensus operator) is minimized by

$$w_i^{n+1} \propto \left[\prod_{k=1}^{N} \left(\sum_{j=1}^{m_k} \tilde{\pi}_{i,j}^{(k,1),n+1}\right)\right]^{1/N}, \quad \sum_{i=1}^{m} w_i^{n+1} = 1 \ . \tag{4.26}$$

However, the above equation is a geometric mean, which is numerically unstable when $\sum_{j=1}^{m_k} \tilde{\pi}_{i,j}^{(k,1),n+1} \to 0^+$ for some combination of i and k. Here, we employ a different technique. Let

$$\tilde{w}_i^{(k,1),n+1} \propto \sum_{j=1}^{m_k} \tilde{\pi}_{i,j}^{(k,1),n+1}, \quad \text{s.t.} \sum_{i=1}^{m} \tilde{w}_i^{(k,1),n+1} = 1.$$

Let the distribution $\tilde{\boldsymbol{w}}^{(k),n+1} = (\tilde{w}_i^{(k,1),n+1})_{i=1,\ldots,m}$. Then Eq. (4.25) is equivalent to $\min_{\boldsymbol{w} \in \Delta_m} \sum_{k=1}^{N} \mathrm{KL}(\boldsymbol{w}, \tilde{\boldsymbol{w}}^{(k),n+1})$. Essentially, a consensus $\boldsymbol{w}$ is sought to minimize the sum of KL divergence. In the same spirit, we propose to find a consensus by changing the order of $\boldsymbol{w}$ and $\tilde{\boldsymbol{w}}^{(k),n+1}$ in the KL divergence:

$\min_{\boldsymbol{w}\in\Delta_m} \sum_{k=1}^{N} \mathrm{KL}(\tilde{\boldsymbol{w}}^{(k),n+1}, \boldsymbol{w}) =$

$$\min_{\boldsymbol{w}\in\Delta_m} \sum_{k=1}^{N}\sum_{i=1}^{m} \tilde{w}_i^{(k,1),n+1}(\log(\tilde{w}_i^{(k,1),n+1}) - \log(w_i)) \ , \tag{4.27}$$

which again has a closed form solution:

$$(\mathrm{R1}) : w_i^{n+1} \propto \frac{1}{N}\sum_{k=1}^{N} \tilde{w}_i^{(k,1),n+1}, \ \ \sum_{i=1}^{m} w_i^{n+1} = 1 \ . \tag{4.28}$$

The solution of Eq. (4.27) overcomes the numerical instability. We will call this heuristic update rule as (R1), which has been employed in the Bregman clustering method [37]. In addition, a slightly different version of update rule can be

$$(\mathrm{R2}) : \left(w_i^{n+1}\right)^{1/2} \propto \frac{1}{N}\sum_{k=1}^{N} \left(\tilde{w}_i^{(k,1),n+1}\right)^{1/2}, \ \ \sum_{i=1}^{m} w_i^{n+1} = 1 \ . \tag{4.29}$$

In Section 4.4, we conduct experiments for testing both (R1) and (R2). We have tried other update rules, such as Fisher-Rao Riemannian center [49], and found that the experimental results do not differ much in terms of the converged objective function. It is worth mentioning that neither (R1) nor (R2) ensures the convergence to a (local) minimum.

We summarize the B-ADMM approach in Algorithm 4.5. The implementation involves one hyper-parameters ρ (by default, $\tau = 10$). In our implementation, we choose ρ relatively according to Eq. (4.14). To the best of our knowledge, the convergence of B-ADMM has not been proved for our formulation (even under fixed support points $\boldsymbol{x}$) although this topic has been pursued in recent literature [31]. In the general case of solving Eq. (4.2), the optimization of the cluster centroid is non-convex because the support points are updated after B-ADMM is applied to optimize the weights. In Section 4.3.7, we empirically test the convergence of the centroid optimization algorithm based on B-ADMM. We found that B-ADMM usually converges quickly to a moderate accuracy, making it preferable for D2-clustering. In our implementation, we use a fixed number of B-ADMM iterations (by default, 100) across multiple assignment-update rounds in D2-clustering.

Algorithm 4.5 Centroid Update with B-ADMM

procedure CENTROID($\{P^{(k)}\}_{k=1}^N$, P, Π).
  $\Lambda := 0$; $\bar{\Pi}^{(2),0} := \Pi$.
  repeat
    Update $\boldsymbol{x}$ from Eq.(4.3) per τ loops;
    for $k = 1, \ldots, N$ **do**
      Update $\Pi^{(k,1)}$ based on Eq.(4.21) (4.22);
      Update $\{\tilde{\pi}_{i,j}^{(k,1)}\}$ based on Eq.(4.23);
    end for
    Update $\boldsymbol{w}$ based on Eq.(4.28) or Eq.(4.29);
    for $k = 1, \ldots, N$ **do**
      Update $\Pi^{(k,2)}$ based on Eq.(4.24);
      $\Lambda^{(k)} := \Lambda^{(k)} + \rho(\Pi^{(k,1)} - \Pi^{(k,2)})$;
    end for
  until P converges
  return P
end procedure

4.3.4 Algorithm Initialization and Implementation

In this section, we explain some specifics in the implementation of the algorithms, such as initialization, warm-start in optimization, measures for further speed-up, and the method for parallelization.

The number of support vectors in the centroid distribution, denoted by m, is set to the average number of support vectors in the distributions in the corresponding cluster. To initialize a centroid, we select randomly a distribution with at least m support vectors from the cluster. If the number of support vectors in the distribution is larger than m, we will merge recursively a pair of support vectors according to an optimal criterion until the support size reaches m, similar to the process used in linkage clustering. Consider a chosen distribution $P = \{(w_1, x_1), ..., (w_m, x_m)\}$. We merge x_i and x_j to $\bar{x} = (w_i x_i + w_j x_j)/\bar{w}$, where $\bar{w} = w_i + w_j$ is the new weight for $\bar{x}$, if (i, j) solves

$$\min_{i,j} w_i w_j \|x_i - x_j\|^2/(w_i + w_j)\,. \tag{4.30}$$

Let the new distribution after one merge be P'. It is sensible to minimize the Wasserstein distance between P and P' to decide which support vectors to merge.

We note that

$$W^2(P, P') \leq w_i \|x_i - \bar{x}\|^2 + w_i \|x_j - \bar{x}\|^2 .$$

This upper bound is obtained by the transport mapping x_i and x_j exclusively to $\bar{x}$ and the other support vectors to themselves. To simplify computation, we instead minimize the upper bound, which is achieved by the $\bar{x}$ given above and by the pair (i, j) specified in Eq. (4.30).

The B-ADMM method requires an initialization for $\Pi^{(k,2)}$, where k is the index for every cluster member, before starting the inner loops (see Algorithm 4.5). We use a warm-start for $\Pi^{(k,2)}$. Specifically, for the members whose cluster labels are unchanged after the most recent label assignment, $\Pi^{(k,2)}$ is initialized by its value solved (and cached) in the previous round (with respect to the outer loop) . Otherwise, we initialize $\Pi^{(k,2)} = (\pi_{i,j}^{(k,2)})$, $i = 1, ..., m$, $j = 1, ..., m_k$ by $\pi_{i,j}^{(k,2),0} := w_i w_j^{(k)}$. This scheme of initialization is also applied in the first round of iteration when class labels are assigned for the first time and there exists no previous solution for this parameter.

At the relabeling step (*i.e.*, to assign data points to centroids after centroids are updated), we need to compute $\bar{N}K$ Wasserstein distances, where $\bar{N}$ is the data size and K is the number of centroids. This part of the computation, usually negligible in the original D2-clustering, is a sizable cost in our new algorithms. To further boost the scalability, we employ the technique of [50] to skip unnecessary distance calculation by exploiting the triangle inequality of a metric.

In our implementation, we use a fixed number of iterations ϵ_i for all inner loops for simplicity. Obtaining highly accurate result for the inner loop is not crucial because the partition will be changed by the outer loop. For B-ADMM, we found that setting ϵ_i to tens or a hundred suffices. For subgradient descent and ADMM, an even smaller ϵ_i is sufficient, *e.g.*, around or below ten. The number of iterations of the outer loop ϵ_o is not fixed, but rather is adaptively determined when a certain termination criterion is met.

With an efficient serial implementation, our algorithms can be deployed to handle moderate scale data on a single PC. We also implemented their parallel versions which are scalable to a large data size and a large number of clusters. We use the commercial solver provided by Mosek,[2] which is among the fastest

[2] https://www.mosek.com

LP/QP solvers available. In particular, Mosek provides optimized simplex solver for transportation problems that fits our needs well.

The algorithms we have developed here are all readily parallelizable by adopting the Allreduce framework in MPI. In our implementation, we divide data evenly into trunks and process each trunk at one processor. Each trunk of data stay at the same processor during the whole program. We can parallelize the algorithms simply by dividing the data because in the centroid update step, the computation comprises mainly separate per data point optimization problems. The main communication cost is on synchronizing the update for centroids by the inner loop. The synchronization time with equally partitioned data is negligible.

We experimented with discrete distributions over a vector space endowed with the Euclidean distance as well as over a symbolic set. In the second case, a symbol-to-symbol distance matrix is provided. When applying D2-clustering to such data, the step of updating the support vectors can be skipped since the set of symbols is fixed. In some datasets, the support vectors in the distributions locate only on a pre-given grid. We can save memory in the implementation by storing the indices of the grid points rather than the direct vector values.

Although we assume each instance is a single distribution in all the previous discussion, it is straightforward to generalize to the case when an instance is an array of distributions (indeed that is the original setup of D2-clustering in [19]). For instance, a protein sequence can be characterized by three histograms over respectively amino acids, dipeptides, and tripeptides. This extension causes little extra work in the algorithms. When updating the cluster centroids, the distributions of different modalities can be processed separately, while in the update of cluster labels, the sum of squared Wasserstein distances for all the distributions is used as the combined distance.

4.3.5 Complexity and Performance Comparisons

Recall some notations: $\bar{N}$ is the data size (total number of distributions to be clustered); d is the dimension of the support vectors; K is the number of clusters; and ϵ_i or ϵ_o is the number of iterations in the inner or outer loop. Let $\bar{m}$ be the average number of support vectors in each distribution in the training set and m be the number of support vectors in each centroid distribution ($\bar{m} = m$ in our setup).

In our implementation, to reduce the time of dynamic memory allocation, we retain the memory for the matching weights between the support vectors of each distribution and its corresponding centroid. Hence, the memory allocation is of order $O(\bar{N}\bar{m}m) + O(d\bar{N}\bar{m} + dKm)$.

For computational complexity, first consider the time for assigning cluster labels in the outer loop. Without the acceleration yielded from the triangle inequality, the complexity is $O(\epsilon_o \bar{N} K l(\bar{m}m, d))$, where $l(\bar{m}m, d)$ is the average time to solve the Wasserstein distance between distributions on a d dimensional metric space. Empirically, we found that by omitting unnecessary distance computation via the triangle inequality, the complexity is reduced roughly to $O(\epsilon_o(\bar{N} + K^2) l(\bar{m}m, d))$. For the centroid update step, the time complexity of the serial version of the ADMM method is $O(\epsilon_o \epsilon_i \bar{N} md) + O(T_{admm} \cdot \epsilon_o \epsilon_i \bar{N} q(\bar{m}m, d))$, where $q(m'm, d)$ is the average time to solve QPs (Eq. (4.12)). The complexity of the serial B-ADMM is $O(\epsilon_o \epsilon_i \bar{N} md/\tau) + O(\epsilon_o \epsilon_i \bar{N} \bar{m} m)$. Note that in the serial algorithms, the complexity for updating centroids does not depend on K, but only on data size $\bar{N}$. For the parallel versions of the algorithms, the communication load per iteration in the inner loop is $O(T_{admm} K m d)$ for ADMM and $O(Km(1 + d/\tau))$ for the B-ADMM.

Both analytical and empirical studies (Section 4.3.8) show that the ADMM algorithm is significantly slower than the other two when the data size is large due to the many constrained QP sub-problems required. Although the theoretical properties of convergence are better understood for ADMM, our experiments show that B-ADMM performs well consistently in terms of both convergence and the quality of the clustering results.

Although the preference for B-ADMM is experimentally validated, given the lack of strong theoretical results on its convergence, it is not clear-cut that B-ADMM can always replace the alternatives. We were thus motivated to develop the subgradient descent (in our supplement) and standard ADMM algorithms to serve at least as yardsticks for comparison. We provide the following guidelines on the usage of the algorithms.

- We recommend the modified B-ADMM as the default data processing pipeline for its scalability, stability, and fast performance. Large memory is assumed to be available under the default setting.

- It is known that ADMM type methods can approach the optimal solution

quickly at the beginning when the current solution is far from the optimum while the convergence slows down substantially when the solution is in the proximity of the optimum. Because we always reset the Lagrangian multipliers in B-ADMM at the beginning of every round of the inner loop and a fixed number of iterations are performed within the loop, our scheme does not pursue aggressively high accuracy for the resulting centroids at every round. However, if the need arises for highly accurate centroids, we recommend the subgradient descent method that takes as initialization the centroids first obtained by B-ADMM.

4.3.6 Experimental Setup

We have conducted experiments to examine the convergence of the algorithms, stability, computational/memory efficiency and scalability of the algorithms, and quality of the clustering results on large data from several domains.

Table 4.1. Datasets in the experiments. $\bar{N}$: data size, d: dimension of the support vectors ("symb" for symbolic data), m: number of support vectors in a centroid, K: maximum number of clusters tested. An entry with the same value as in the previous row is indicated by "-".

Data	$\bar{N}$	d	m	K
synthetic	2,560,000	$\geq$16	$\geq$32	256
image color	5,000	3	8	10
image texture	-	-	-	-
protein sequence 1-gram	10,742	symb.	20	10
protein sequence 3-gram	-	-	32	-
USPS digits	11,000	2	80	360
BBC news abstract	2,225	300	16	15
Wiki events abstract	1,983	400	16	100
20newsgroups GV	18,774	300	64	40
20newsgroups WV	-	400	100	-

Table 4.1 lists the basic information about the datasets used in our experiments. For the synthetic data, the support vectors are generated by sampling from a multivariate normal distribution and then adding a heavy-tailed noise from the student's t-distribution. The probabilities on the support vectors are perturbed

and normalized samples from Dirichlet distribution with symmetric prior. We omit details due to lack of space. The synthetic data are only used to study the scalability of the algorithms. The image color or texture data are created from crawled general-purpose photographs. Local color or texture features around each pixel in an image are clustered (*i.e.* quantized) to yield color or texture distributions. The protein sequence data are histograms over the amino acids (1-gram) and tripeptides (3-tuples, 3-gram) [38]. The USPS digit images are treated as normalized histograms over the pixel locations covered by the digits, where the support vector is the 2D coordinate of a pixel and the weight corresponds to pixel intensity. For the *20newsgroups* data, we use the recommended "bydate" MATLAB version which includes 18,774 documents and 61,188 unique words. The two datasets, "20 newsgroup GV" and "20newsgroup WV" are created by characterizing the documents in different ways. The "BBC news abstract" and "Wiki events abstract" datasets are truncated versions of two document collections [51,52]. These two sets of short documents retain only the title and the first sentence of each original post. The purpose of using these severely cut documents is to investigate a more challenging setting for existing document or sentence analysis methods, where semantically related sentences are less likely to share the exact same words. For example, "NASA revealed its ambitions that humans can set foot on Mars" and "US is planning to send American astronauts to Red Planet" describe the same event. More details on the data are referred to Section 4.3.9.

4.3.7 Convergence and Stability

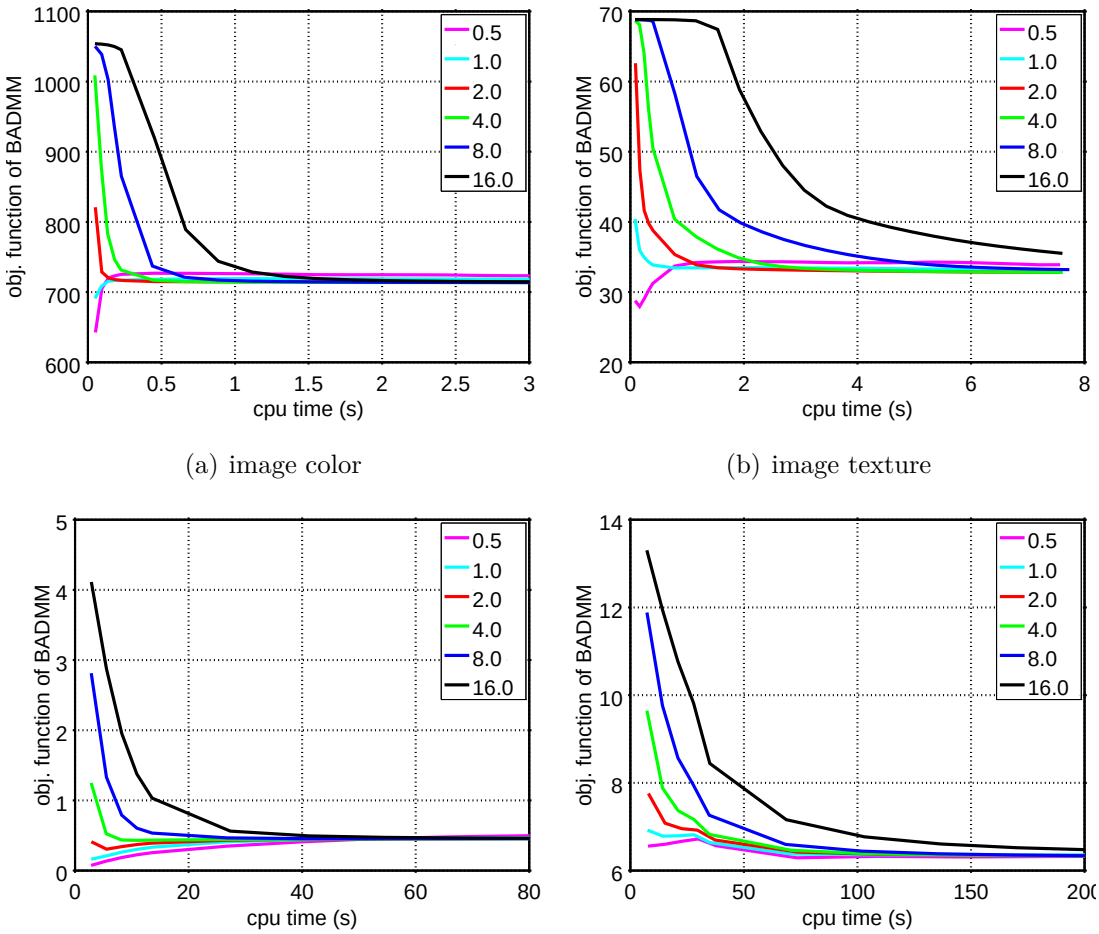
We empirically test the convergence and stability of the three approaches: modified B-ADMM, ADMM, and subgradient descent method, based on their sequential versions implemented in the C programming language. Four datasets are used in the test: protein sequence 1-gram, 3-gram data, and the image color and texture data. In summary, the experiments show that the modified B-ADMM method has achieved the best numerical stability with respect to hyper-parameters while keeping a comparable convergence rate as the subgradient descent method in terms of CPU time. To conserve space, detailed results on the study of stability are provided in Appendix B. Despite of its popularity in large-scale machine learning problems, by lifting $\bar{N}$ LPs to $\bar{N}$ QPs, the ADMM approach is much slower on

large datasets than the other two approaches are.

We examine the convergence property of the B-ADMM approach for computing the centroid of a single cluster (the inner loop). In this experiment, a subset of image color or texture data with size $2,000$ is used. For the two protein sequence datasets, the whole sets are used. Fig. 4.1 and Fig. 4.2 show the convergence analysis results on the four datasets. The vertical axis in the plots in Fig. 4.1 is the objective function of B-ADMM, given in Eq. (4.18), but not the original objective function of clustering in Eq. (4.2). The runtime is based on a single thread with 2.6 GHz Intel Core i7. The plots reveal two characteristics about the B-ADMM approach: 1) The algorithm achieves consistent and comparable convergence rate under a wide range of values for the hyper-parameter $\rho_0 \in \{0.5, 1.0, 2.0, 4.0, 8.0, 16.0\}$ and is numerically stable; 2) The effect of the hyper-parameter on the decreasing ratio of the dual and primal residuals follows similar patterns across the datasets.

It is technically subtle to compare the convergence and stability of the overall AD2-clustering embedded with different algorithms for computing the centroid. Because of the many iterations in the outer loop, the centroid computation algorithm (solving the inner loop) may behave quite differently over the outer-loop rounds. For instance, if an algorithm is highly sensitive to a hyper-parameter in optimization, the hyper-parameter chosen based on earlier rounds may yield slow convergence later or even cause the failure of convergence. Moreover, achieving high accuracy for centroids in earlier rounds, usually demanding more inner-loop iterations, may not necessarily result in faster decrease in the clustering objective function because the cluster assignment step also matters.

In light of these issues, we employ a protocol described in Algorithm 4.6 to decide the number of iterations in the inner loop. The protocol specifies that in each iteration of the outer loop, the inner loop for updating centroids should complete within $\eta T_a/K$ amount of time, where T_a is the time used by the assignment step and K is the number of clusters. As we have pointed out, the LP/QP solver in the subgradient descent method or standard ADMM suffers from rapidly increasing complexity when the number of support points per distribution increases. In contrast, the effect on B-ADMM is much lower. In the experiment below, the datasets contain distributions with relatively small support sizes (a setup favoring the former two methods). A relatively tight time-allocation $\eta = 2.0$ is set. The subgradient descent method finishes at most 2 iterations in the inner loop, while



(a) image color

(b) image texture

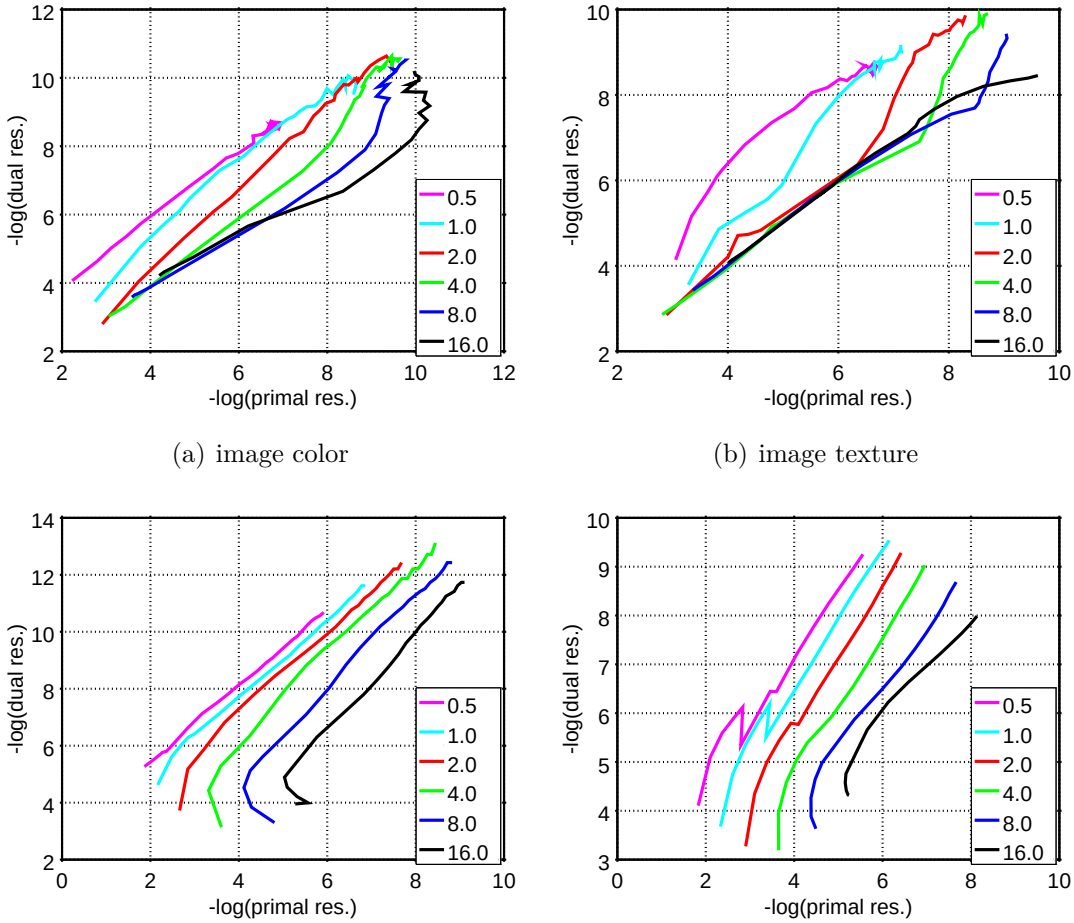
(c) prot. seq. 1-gram

(d) prot. seq. 3-gram

Figure 4.1. Convergence analysis of the B-ADMM method for computing a single centroid based on four datasets: objective function of B-ADMM based centroid computation with respect to CPU time.

B-ADMM on average finishes more than 60 iterations on the color and texture data, and more than 20 iterations on the protein sequence 1-gram and 3-gram data. The results by the ADMM method are omitted because this method cannot finish a single iteration under this time allocation.

In Fig. 4.3, we compare the convergence performance of the overall clustering process employing B-ADMM at $\rho_0 = 2.0$ and the subgradient descent method with fine-tuned values for the step-size parameter $\alpha \in \{0.05, 0.1, 0.25, 0.5, 1.0, 2.0, 5.0, 10.0\}$. The step-size is chosen as the value yielding the lowest clustering objective function in the first round. In this experiment, the whole image color and texture data



(a) image color

(b) image texture

(c) prot. seq. 1-gram

(d) prot. seq. 3-gram

Figure 4.2. Convergence analysis of the B-ADMM method for computing a single centroid based on four datasets: the trajectory of dual residual vs. primal residual (in the negative log scale.

are used. In the plots, the clustering objective function (Eq. (4.2)) is shown with respect to the CPU time. We observe a couple of advantages of the B-ADMM method. First, with a fixed parameter ρ_0, B-ADMM yields good convergence on all the four datasets, while the subgradient descent method requires manually tuning the step-size α in order to achieve comparable convergence speed. Second, B-ADMM achieves consistently lower values for the objective function across time. On the protein sequence 1-gram data, B-ADMM converges substantially faster than the subgradient descent method with a fine-tuned step-size. Moreover, the subgradient descent method is numerically less stable. Although the step-size is

Algorithm 4.6 Time allocation based algorithmic profile protocol

procedure PROFILE($\{P^{(k)}\}_{k=1}^{M}$, Q, K, η).
    Start profiling;
    $T = 0$;
    repeat
        $T_a = 0$;
        Assignment Step;
        GetElaspedCPUTime(T_a, T);
        GetAndDisplayPerplexity(T);
        Update Step within CPU time allocation $\eta T_a/K$;
    until $T < T_{total}$
    return
end procedure

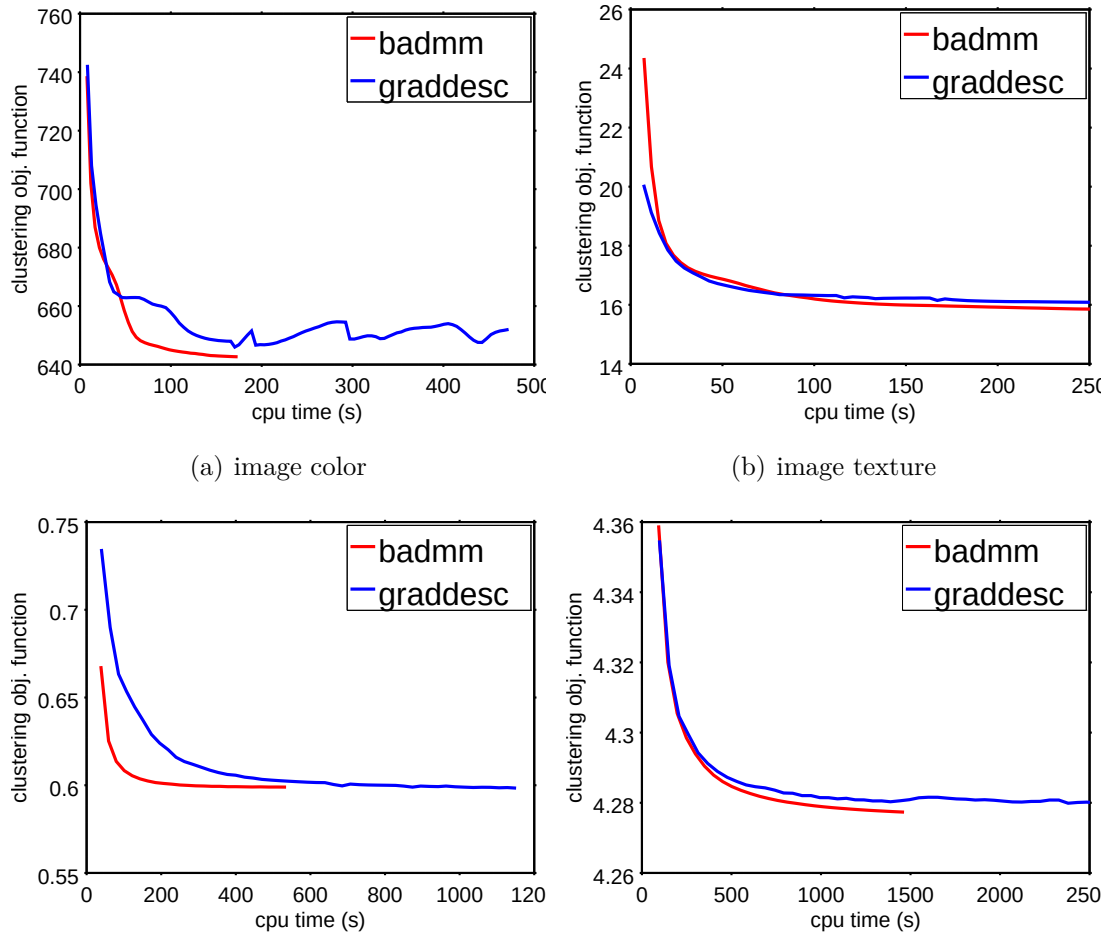
fine-tuned based on the performance at the beginning, on the image color data, the objective function fluctuates noticeably in later rounds. Striking a balance (assuming it exists) between stability and speed for the subgradient descent method is a difficult dilemma.

4.3.8 Efficiency and Scalability

We now study the computational/memory efficiency and scalability of AD2-clustering with the B-ADMM algorithm embedded for computing cluster centroids. We use the synthetic data that allow easy control over data size and other parameters in order to test their effects on the computational and memory load (*i.e.*, workload) of the algorithm. We study the *scalability* of our parallel implementation on a cluster computer with distributed memory. Scalability here refers to the ability of a parallel system to utilize an increasing number of processors.

AD2-clustering can be both CPU-bound and memory-bound. Based on the observations from the above serial experiments, we conducted three sets of experiments to test the scalability of AD2-clustering in a multi-core environment, specifically, strong scaling efficiency, weak scaling efficiency with respect to $\bar{N}$ or m. The configuration ensures that each iteration finishes within one hour and the memory of a typical computer cluster is sufficient.

Strong scaling efficiency (SSE) is about the speed-up gained from using more and more processors when the problem is fixed in size. Ideally, the runtime on parallel CPUs is the time on a single thread divided by the number of CPUs. In



(a) image color

(b) image texture

(c) prot. seq. 1-gram

(d) prot. seq. 3-gram

Figure 4.3. Convergence performance of B-ADMM and the subgradient descent method for D2-clustering based on four datasets. The clustering objective function versus CPU time is shown. Here, $K = 10$, and the time-allocation ratio $\eta = 2.0$.

Table 4.2. Scaling efficiency of AD2-clustering in parallel implementation.

# processors	32	64	128	256	512
SSE (%)	93.9	93.4	92.9	84.8	84.1
WSE on $\bar{N}$ (%)	99	94.8	95.7	93.3	93.2
WSE on m (%)	96.6	89.4	83.5	79.0	-

practice, such a reduction in time cannot be fully achieved due to communication between CPUs and time for synchronization. We thus measure SSE by the ratio between the ideal and the actual amount of time. We chose a moderate size problem that can fit in the memory of a single machine (50GB): $\bar{N} = 250000$,

$d = 16$, $m = 64$, $k = 16$. Table 4.2 shows the SSE values with the number of processors ranging from 32 to 512. The results show that AD2-clustering scales well in SSE when the number of processors is up to hundreds.

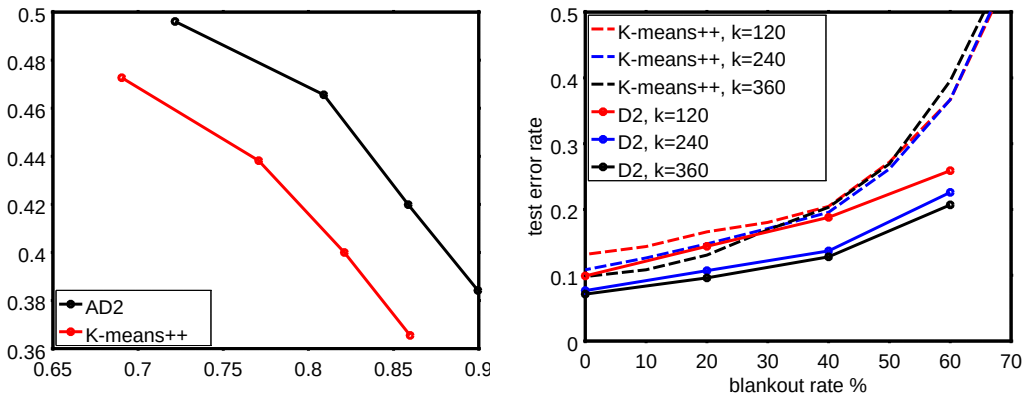
Weak scaling efficiency (WSE) measures how stable the real computation time can be when proportionally more processors are used as the size of the problem grows. We compute WSE with respect to both $\bar{N}$ and m. Let np be the number of processors. For WSE on $\bar{N}$, we set $\bar{N} = 5000 \cdot np$, $d = 64$, $m = 64$, and $K = 64$ on each processor. The per-node memory is roughly $1GB$. For WSE on m, we set $\bar{N} = 10000$, $K = 64$, $d = 64$, and $m = 32 \cdot \sqrt{np}$. Table 4.2 shows the values of WSE on $\bar{N}$ and m. We can see that AD2-clustering also has good weak scalability, making it suitable for handling large scale data. In summary, our proposed method can be effectively accelerated with an increasing number of CPUs.

4.3.9 Quality of Clustering Results

Handwritten Digits: We conducted two experiments to evaluate the results of AD2-clustering on USPS data, which contain 1100×10 instances ($1,100$ per class). First, we cluster the images at $K = 30, 60, 120, 240$ and report in Fig. 4.4(a) the homogeneity versus completeness [53] of the obtained clustering results. We set K to large values because clustering performed on such image data is often for the purpose of quantization where the number of clusters is much larger than the number of classes. In this case, homogeneity and completeness are more meaningful measures than the others used in the literature (several of which will be used later for the next two datasets). Roughly speaking, completeness measures how likely members of the same true class fall into the same cluster, while homogeneity measures how likely members of the same cluster belong to the same true class. By construction, the two measures have to be traded off. We compared our method with Kmeans++ [54]. For this dataset, we found that Kmeans++, with more careful initialization, yields better results than the standard K-means. Their difference on the other datasets is negligible. Fig. 4.4(a) shows that AD2-clustering obviously outperforms Kmeans++ cross K's.

Secondly, we tested AD2-clustering for quantization with the existence of noise. In this experiment, we corrupted each sample by "blankout"—randomly deleting a percentage of pixels occupied by the digit (setting to zero the weights of the

corresponding bins), as is done in [55]. Then each class is randomly split into 800/300 training and test samples. Clustering is performed on the 8000 training samples; and a class label is assigned to each cluster by majority vote. In the testing phase, classifying an instance entails locating its nearest centroid and then assigning the class label of the corresponding cluster. The test classification error rates with respect to K and the blankout rate are plotted in Fig. 4.4(b). The comparison with Kmeans++ demonstrates that AD2-clustering performs consistently better, and the margin is remarkable when the number of clusters is large and the blankout rate is high.



(a) Homogeneity vs. completeness (b) Test error rate vs. blankout rate

Figure 4.4. Comparisons between Kmeans++ and AD2-clustering on USPS dataset. We empirically set the number of support vectors in the centroids $m = 80(1 - blankout\_rate)$.

Documents as Bags of Word-vectors: The idea of treating each document as a bag of vectors has been explored in previous work where a nearest neighbor classifier is constructed using Wasserstein distance [5, 56]. One advantage of the Wasserstein distance is to account for the many-to-many mapping between two sets of words. However, clustering based on Wasserstein distance, especially the use of Wasserstein barycenter, has not been explored in the literature of document analysis. We have designed two kinds of experiments using different document data to assess the power of AD2-clustering.

To demonstrate the robustness of D2-clustering across different word embedding spaces, we use 20newsgroups processed based on two pre-trained word embedding models. We pre-processed the dataset by two steps: remove stop words and remove other words that do not belong to a pre-selected background vocabulary. In

GV Vocab.	tf-idf	LDA	LDA naïve	Avg. vector	AD2	AD2	AD2
K	40	20	20	30	20	30	40
AMI	0.447	0.326	0.329	0.360	0.418	**0.461**	0.446
ARI	0.151	0.160	0.187	0.198	0.260	0.281	**0.284**
hours					5.8	7.5	10.4
# iter.					44	45	61
WV Vocab.	tf-idf	LDA	LDA naïve	Avg. vector	AD2	AD2	AD2
K	20	25	20	20	20	30	40
AMI	0.432	0.336	0.345	0.398	0.476	**0.477**	0.455
ARI	0.146	0.164	0.183	0.212	**0.289**	0.274	0.278
hours					10.0	11.3	17.1
# iter.					28	29	36

Table 4.3. Compare clustering results of AD2-clustering and several baseline methods using two versions of Bag-of-Words representation for the 20newsgroups data. Top panel: the data are extracted using the GV vocabulary; bottom panel: WV vocabulary. AD2-clustering is performed once on 16 cores with less than 5GB memory. Run-times of AD2-clustering are reported (along with the total number of iterations).

particular, two background vocabularies are tested: English Gigaword-5 (denoted by GV) [57] and a Wikipedia dump with minimum word count of 10 (denoted by WV) [58]. Omitting details due to lack of space, we validated that under the GV or WV vocabulary information relevant to the class identities of the documents is almost intact. The words in a document are then mapped to a vector space. For GV vocabulary, the Glove mapping to a vector space of dimension 300 is used [57], while for WV, the Skip-gram model is used to train a mapping space of dimension 400 [58]. The frequencies on the words are adjusted by the popular scheme of tf-idf. The number of different words in a document is bounded by m (its value in Table 4.1). If a document has more than m different words, some words are merged into hyper-words recursively until reaching m, in the same manner as the greedy merging scheme used in centroid initialization described in Section 4.3.4.

We evaluate the clustering performance by two widely used metrics: AMI [6] and ARI [59, 60]. The baseline methods for comparison include K-means on the raw tf-idf word frequencies, K-means on the LDA topic proportional vectors [61] (the

number of LDA topics is chosen from $\{40, 60, 80, 100\}$), K-means on the average word vectors, and the naïve way of treating the 20 LDA topics as clusters. For each baseline method, we tested the number of clusters $K \in \{10, 15, 20, 25, 30, 40\}$ and report only the best performance for the baseline methods in Table 4.3, while for AD2-clustering, $K = 20, 30, 40$ are reported. Under any given setup of a baseline method, multiple runs were conducted with different initialization and the median value of the results was taken. The experimental results show that AD2-clustering achieves the best performance on the two datasets according to both AMI and ARI. Comparing with most baseline methods, the boost in performance by AD2-clustering is substantial. Furthermore, we also vary m in the experimental setup of AD2-clustering. At $m = 1$, our method is exactly equivalent to K-means of the distribution means. We increased m empirically to see how the results improve with a larger m. We did not observe any further performance improvement for $m \geq 64$.

We note that the high competitiveness of AD2-clustering can be credited to (1) a reasonable word embedding model and (2) the bag-of-words model. When the occurrence of words is sparse across documents, the semantic relatedness between different words and their compositions in a document plays a critical role in measuring the document similarity.

In our next experiment, we study AD2-clustering for short documents, a challenging setting for almost all existing methods based on the bag-of-words representation. The results show that the performance boost of AD2-clustering is also substantial. We use two datasets, one is called "BBC news abstract" and the other "Wiki events abstract". Each document is represented by only the title and the first sentence from a news article or an event description. Their word embedding models are same as the one used by the "WV" version in our previous experiment. The "BBC news" dataset contains five news categories, and "Wiki events" dataset contains 54 events. Clustering such short documents is more challenging due to the sparse nature of word occurrences. As shown by Table 4.4, in terms of generating clusters coherent with the labeled categories or events, methods which leverage either the bag-of-words model or the word embedding model (but not both) are outperformed by AD2-clustering which exploits both. In addition, AD2-clustering is fast for those sparse support discrete distribution data. It takes only several minutes to finish the clustering in an 8-core machine.

	Tf-idf	LDA	NMF	Avg. vector	AD2
BBC news abstract	0.376	0.151	0.537	0.753	0.759
Wiki events abstract	0.448	0.280	0.395	0.312	0.545

Table 4.4. Best AMIs achieved by different methods on the two short document datasets. NMF denotes for the non-negative matrix factorization method.

To quantify the gain from employing an effective word embedding model, we also applied AD2-clustering to a random word embedding model, where a vector sampled from a multivariate Gaussian with 300 dimensions is used to represent a word in vocabulary. We found that the results are much worse than those reported in Table 4.4 for AD2-clustering. The best AMI for "BBC news abstract" is 0.187 and the best AMI for "Wiki events abstract" is 0.369, comparing respectively with 0.759 and 0.545 obtained from a carefully trained word embedding model.

4.4 Discussions

Both the B-ADMM and IBP can be rephrased into two-step iterative algorithms via mirror maps (in a similar way of mirror prox [62] or mirror descent [63]). One step is the free-space move in the dual space, and the other is the Bregman projection (as used in IPFP) in the primal space. Let $\Phi(\cdot)$ be the entropy function, the mirror map used by IBP is $\Phi(\Pi)$, while the mirror map of B-ADMM is

$$\Phi(\Pi^{(1)}, \Pi^{(2)}, \Lambda) = \Phi(\Pi^{(1)}) + \Phi(\Pi^{(2)}) + \frac{\|\Lambda\|^2}{\rho^2},$$

where Λ is the dual coordinate derived from relaxing constraints $\Pi^{(1)} = \Pi^{(2)}$ to a saddle point reformulation. IBP alternates the move $-\left[\frac{C}{\varepsilon}\right]$ in the dual space and projection in Δ_1 or Δ_2 of the primal space. In comparison, B-ADMM alternates the move

$$-\begin{bmatrix} \frac{C+\Lambda}{\rho} + \nabla\Phi(\Pi^{(1)}) - \nabla\Phi(\Pi^{(2)}) \\ -\frac{\Lambda}{\rho} + \nabla\Phi(\Pi^{(2)}) - \nabla\Phi(\Pi^{(1)}) \\ \rho(\Pi^{(1)} - \Pi^{(2)}) \end{bmatrix},$$

and the projection in $\Delta_1 \times \Delta_2 \times \mathbb{R}_{m_1 \times m_2}$.[3] The convergence of B-ADMM is not evident from the conventional optimization literature [64] because the move is not monotonic (It is still monotonic for standard ADMM). We conduct two pilot studies to compare B-ADMM and IBP in terms of convergence behavior and the quality of the Wasserstein barycenters with a sparse finite support set, where quality is measured by the objective function achieved.

Although the second pilot study shows certain advantages of B-ADMM, we clarify that the study is not intended to demonstrate which algorithm is better in the general context of the OT problem. In fact, for the OT problem alone, IBP is more solid in theory than B-ADMM in two aspects. First, IBP has linear convergence, while B-ADMM only has a sub-linear rate [31]. Second, IBP yields an OT solution more accurately satisfying the coupling constraints than B-ADMM can offer with the same computational time. In the Wasserstein barycenter problem we tackle here, either algorithm must be embedded into an outer loop, which calls for practical considerations other than solving a stand-alone OT problem. We will elaborate on these points below.

Benamou *et al.* [21], in their Figure 1, show how their algorithm progressively shifts mass away from the diagonal over the iterations. We adopt the same study here and visualize in Fig. 3.1 how the mass transport between two 1D distributions evolves over iterations. As a qualitative study, we observe that the entropy regularization term in IBP introduces clearly smoothing effect on the final solutions. It has been pointed out in the literature that the smoothing effect of entropy regularization diminishes as parameter ε decreases. In our study, we found the smoothing effect is also affected by the support size of a distribution. A smooth distribution with large support tends to have higher entropy, thus a relatively smaller ε is needed to achieve similar results. Fig. 3.1 shows that at $\varepsilon = 0.1/N$, the mass transport of IBP at 5,000 iterations is close to the unregularized solution albeit with noticeable difference. Setting ε even smaller (say $0.04/N$) introduces double-precision overflow. As suggested by one of the reviewers, this numerical difficulty can be addressed by thresholding entries that are too small. We applied the reviewer's suggested IBP code with this technique implemented and obtained an incorrect coupling result with artifacts unexplainable by smoothing, *e.g.*, a non-zero region separated from the correct non-zero region. Yet more recently, we learned

[3] The update is done in the Gauss-Seidel type, not in the usual Jacobi type.

from the reviewer that active research on the thresholding technique is currently underway with new manuscripts emerging while we approached the final revision of this paper. In particular, log domain scaling or more sophisticated schemes have been proposed to stabilize the low ε case for Sinkhorn algorithm [65, 66]. At extra computational costs, these new methods produce sharper coupling results than the standard IBP does. Investigating the effectiveness of those algorithms for the Wasserstein barycenter problem is an interesting future work. In contrast, the output of mass transport by B-ADMM ($\rho_0 = 2$, the default setting) at 5,000 iteration is nearly indiscernible from the unregularized solution by LP. This example suggests that if the smoothing effect on the final coupling is to be avoided, B-ADMM may be preferred over IBP with $\epsilon \to 0$, albeit at a cost of increased computation time. With our implementation of B-ADMM and the IBP code provided by the reviewer, we found that IBP is 10 to 15 times faster per iteration.

Because B-ADMM does not require $\rho_0 \to 0$, no numerical difficulty has been encountered in practice. In fact, the convergence speed of B-ADMM is proven to be independent with ρ_0 [31]. As for the Wasserstein barycenter problem, the minimal required tuning makes embedding B-ADMM in an outer loop easy. Putting the IBP in the outer loop reduces its speed advantage. In existing machine learning practice as well as with our algorithm here, pre-converged solutions of IBP or B-ADMM are used, making the use of a very small ϵ in IBP unnecessary. In our algorithm, however, B-ADMM per iteration slows the process as does updating barycenter support points in high dimensions. Thus, the computational gain from replacing B-ADMM by IBP is clipped by the notable proportion of time required to update the support points. In addition, although the coupling weights solved by B-ADMM do not satisfy the constraints as well as those by IBP do, they are auxiliary variables, while the barycenters are primary. In summary, the edge of either IBP or B-ADMM over the other subsides when they are embedded in our algorithm. In order to quantitatively compare the methods by measures most pertinent to the users, we examine the objective function and computation time in the second pilot study below.

For the second pilot study, we generated a set of 1,000 discrete distributions, each with a sparse finite support set obtained by clustering pixel colors of images [19] ($d = 3$). The average number of support points is around 6. Starting from the same initial estimate, we calculate the approximate Wasserstein barycenter by different

Solved Wasserstein barycenter with a pre-fixed support $(m = 6)/(m = 60)$			
method	iterations	seconds	obj.
full LP	NA	-	834.1 / 709.6
Our approach (R1)	500 / 800	0.78 / 13.0	838.6 / 713.4
Our approach (R2)	400 / 700	0.58 / 11.8	835.5 / 712.3
IBP[19] $\varepsilon_0 = 0.2$	150 / 40	0.06 / 0.11	978.3 / 1073.5
IBP $\varepsilon_0 = 0.1$	620 / 80	0.16 / 0.23	965.9 / 1051.6
IBP $\varepsilon_0 = 0.02$	6,640 / 760	1.50 / 1.49	957.1 / 1039.4
IBP $\varepsilon_0 = 0.01$	17,420 / 9,460	4.17 / 17.0	960.2 / 2343.8*
IBP $\varepsilon_0 = 0.005$	36,270 / 5,110	8.88 / 9.83	1345.1* / 7112.2*

Solved Wasserstein barycenter with an optimized support $(m = 6)/(m = 60)$			
method	iterations	seconds	obj.
full LP	20	-	717.8 / 692.3
Our approach (R1)	2,000	2.91 / 31.1	723.3 / 692.6
Our approach (R2)	2,000	3.02 / 32.2	722.7 / 692.5
IBP $\varepsilon_0 = 0.2$	190 / 60	0.06 / 0.19	733.7 / 703.5
IBP $\varepsilon_0 = 0.1$	730 / 130	0.22 / 0.31	734.4 / 699.5
IBP $\varepsilon_0 = 0.02$	11,860 / 1,590	2.67 / 2.73	734.9 / 705.5
IBP $\varepsilon_0 = 0.01$	33,940 / 5,130	7.71 / 9.01	734.9 / 708.3
IBP $\varepsilon_0 = 0.005$	69,860 / 16,910	16.5 / 30.87	736.1 / 708.6

Table 4.5. Comparing the solutions of the Wasserstein barycenter by LP, modified B-ADMM (our approach) and IBP. The runtime reported is based on MATLAB implementations.

methods. We obtained results for two cases: barycenters with support size $m = 6$ and barycenters with support size $m = 60$. We use relatively small values of m here in comparison with the existing applications of IBP in imaging science because our focus is on large data size but low support size (sparse support). The obtained barycenters are then evaluated by comparing the objective function (Eq. (4.2)) with that solved directly by the full batch LP (Algorithm 4.2)—the yardstick in the comparison.

The bare form of IBP treats the case of pre-fixed support points while B-ADMM does not constrain the locations. In order to compare the two methods on a common ground, we used two tracks of experiments. In the first track the locations of support points in the barycenters are fixed and shared by all the algorithms, while in the second track both locations and weights are optimized. To adopt IBP in the second track, we experimented with a version of IBP that can automatically restart and update its support points. The restart criterion is that the constraint satisfies certain conditions described in [2, 21]. Generally speaking, the larger ε_0 is the fewer iterations are needed before a restart is evoked. This variant of IBP does not have a descending objective.[4] Therefore, we chose to terminate the iterations when $\langle C, \Pi \rangle$ is detected to increase. The actual implementation can be found in our supplement. In MATLAB, we found IBP to be approximately 10 times faster than B-ADMM per iteration. On the other hand, being faster in one iteration does not necessarily mean faster speed overall. We report the exact numbers of iterations and seconds before reaching a stopping criterion for both methods.

The results of the two tracks of experiments for support size $m = 6, 60$ by LP, the modified B-ADMM (R1 and R2 as explained in Section 4.3.3) and IBP, are presented in Table 4.5. The performance is measured by the value achieved for the objective (a.k.a. distortion) function. The results show that in both tracks, when dealing with sparse support distribution data, B-ADMM achieves lower distortion than IBP does, and furthermore, B-ADMM is quite close to LP. The gap between B-ADMM and LP is smaller than the gap between IBP and B-ADMM. On the other hand, when ϵ_0 is relatively large, IBP can be much faster. There is no tuning of hyper-parameters in B-ADMM. For IBP, ε_0 influences the result, but not by too much as long as it is not too small. Considering the fast speed at large ε_0, we may favor relatively large ε_0 when applying IBP. For the first track experiments, the objective function obtained by IBP is considerably larger than that B-ADMM obtains. We could not push the objective function by IBP to the same level of B-ADMM by letting $\varepsilon_0 \to 0$. At $\varepsilon_0 = 0.01, 0.005$, because double-precision overflow occurs, triggering the thresholding trick. The IBP results with thresholding triggered are marked by star (*) in Table 4.5. These results are actually much worse than the others, an observation consistent with the incorrect coupling weights

[4] It is because the entropic regularization term is not considered in the update of support points.

obtained when this trick is applied in the first pilot study. For the second track experiments, B-ADMM still achieves lower values of the objective function, but the difference from IBP is not as remarkable.

In comparison to the full LP approach, the modified B-ADMM does not yield the exact local minimum. This result reminds us that the modified B-ADMM is still an approximation method, and it cannot fully replace subgradient descent based methods that minimize the objective to a true local minimum.

4.5 Wasserstein Non-negative Matrix Factorization

We now illustrate how the proposed Gibbs-OT in Chapter 3 can be used as a ready-to-plugin inexact oracle for a typical WLM — Wasserstein NMF [23,24]. The data parallelization of this framework is natural because the Gibbs-OT samplers subject to different instances are independent.

4.5.1 Problem Formulation

Given a set of discrete probability measures $\{\Phi_i\}_{i=1}^n$ (data) over $\mathbb{R}^d$, we want to estimate a model $\Theta = \{\Psi_k\}_{k=1}^K$, such that for each Φ_i, there exists a membership vector $\beta^{(i)} \in \Delta_K$: $\Phi_i \approx \sum_{k=1}^K \beta_k^{(i)} \Psi_k$, where each Ψ_k is again a discrete probability measure to be estimated. Therefore, Wasserstein NMF reads $\min_{\Theta,\Xi} \sum_{i=1}^n W\left(\Phi_i, \sum_{k=1}^K \beta_k^{(i)} \Psi_k\right)$, where $\Xi = (\beta^{(1)}, \ldots, \beta^{(n)})$ is the collection of membership vectors, and W is the Wasserstein distance. One can write the problem by plugging Eq. (3.3) in the dual formulation:

$$\min_{\Theta,\Xi} \max_{F=\{\mathbf{f}_i\}_{i=1}^n} \quad \sum_{i=1}^n \left[\langle \widehat{\mathbf{w}}^{(i)}, \mathbf{g}_i \rangle - \langle \mathbf{w}^{(i)}, \mathbf{h}_i \rangle\right] \tag{4.31}$$

$$\text{s.t.} \quad \Psi_k = \sum_{i=1}^m v_i^{(k)} \delta_{\mathbf{x}_i} \ , \tag{4.32}$$

$$\widehat{\Phi}^{(i)} = \sum_{k=1}^K \beta_k^{(i)} \Psi_k \ , \tag{4.33}$$

$$\mathbf{f}_i \in \Omega\left(M(\widehat{\Phi}^{(i)}, \Phi_i)\right) \ , \tag{4.34}$$

The work presented in this section has been published in the form of a research paper: Jianbo Ye, James Z. Wang and Jia Li, "A Simulated Annealing based Inexact Oracle for Wasserstein Loss Minimization," *Proceedings of the 34th International Conference on Machine Learning (ICML)*, Sydney, Australia, Vol 70, pp 3940–3948, August 2017.

where $\widehat{\mathbf{w}}^{(i)} \in \Delta_m$ is the weight vector of discrete probability measure $\widehat{\Phi}^{(i)}$ and $\mathbf{w}^{(i)} \in \Delta_{m_i}$ is the weight vector of $\Phi^{(i)}$. $M(\cdot,\cdot)$ denotes the transportation cost matrix between the supports of two measures. The global optimization solves all three sets of variables (Θ, Ξ, F). In the sequel, we assume support points of $\{\Psi_k\}_{k=1}^m$ — $\{\mathbf{x}_i\}_{i=1}^m$ are shared and pre-fixed.

4.5.2 Algorithm

At every epoch, one updates variables either sequentially (indexed by i) or all together. It is done by first executing the Gibbs-OT oracle subject to the i-th instance and then updating $\mathbf{v}^{(k)}$ and the membership vector $\beta^{(i)}$ accordingly at a chosen step size $\gamma > 0$. At the end of each epoch, the temperature parameter T is adjusted $T := T\left(1 - \sqrt{\frac{1}{m+\bar{m}}}\right)$, where $\bar{m} = \frac{1}{n}\sum_{i=1}^n m_i$. For each instance i, the algorithm proceeds with the following steps iteratively:

1. Initiate from the last computed $\mathbf{U}/\mathbf{V}$ sample subject to instance i, execute the Gibbs-OT Gibbs sampler at constant temperature T until a mixing criterion is met, and get $\mathbf{U}_i$.

2. For $k = 1, \ldots, K$, update $\mathbf{v}^{(k)} \in \Delta_m$ based on gradient $\beta_k^{(i)}\mathbf{U}_i$ using the iterates of online mirror descent (MD) subject to the step-size γ [63].

3. Also update the membership vector $\beta^{(i)} \in \Delta_K$ based on gradient

$$(\langle \mathbf{v}^{(1)}, \mathbf{U}_i \rangle, \ldots, \langle \mathbf{v}^{(K)}, \mathbf{U}_i \rangle)^T$$

   using the iterates of accelerated mirror descent (AMD) with restarts subject to the same step-size γ [67].

We note that the practical speed-ups we achieved via the above procedure is the warm-start feature in Step 1. If one uses a black-box OT solver, this dimension of speed-ups is not viable.

4.5.3 Results

We investigate the empirical convergence of the proposed Wasserstein NMF method by two datasets: one is a subset of MNIST handwritten digit images which contains

200 digits of "5", and the other is the ORL 400-face dataset. Our results are based on a C/C++ implementation with vectorization. In particular, we set $K = 40, \gamma = 2.0$ for both datasets. The learned components are visualized together with alternative approaches (smoothed W-NMF [23] and regular NMF) in Appendix Figs. 4.5 and 4.6. From these figures, we observe that our learned components using Gibbs-OT are shaper than the smoothed W-NMF. This can be explained by the fact that Gibbs-OT can potentially push for higher quality of approximation by gradually annealing the temperature. We also observe that the learned components might possess some salt-and-pepper noise. This is because the Wasserstein distance by definition is not very sensitive to the sub-pixel displacements. On a single-core of a 3.3 GHz Intel Core i5 CPU, the average time spent for each epoch for these two datasets are 0.84 seconds and 16.8 seconds, respectively. It is about two magnitude faster than fully solving all OTs via a commercial LP solver [5].

[5]We use the specialized network flow solver in Mosek (https://www.mosek.com) for the computation, which is found faster than general simplex or IPM solver at moderate problem scale.

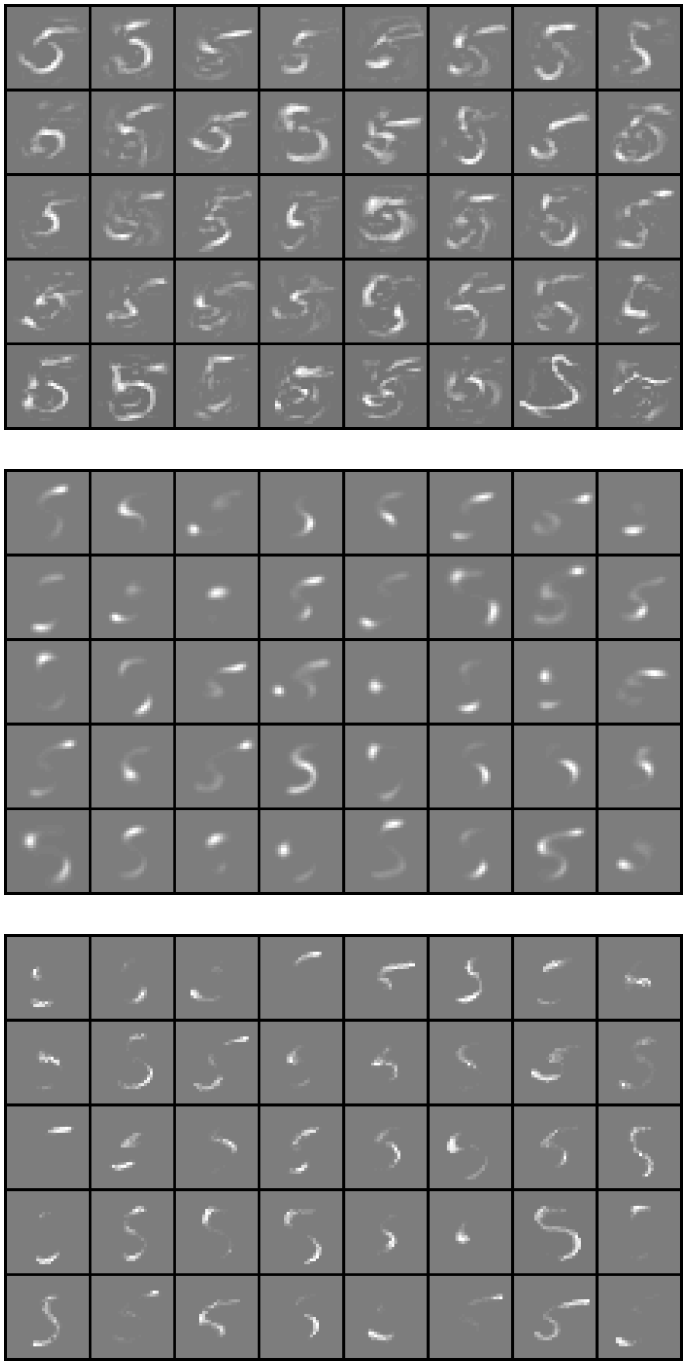


Figure 4.5. NMF components learned by different methods ($K = 40$) on the 200 digit "5" images. Top: regular NMF; Middle: W-NMF with entropic regularization ($\varepsilon = 1/100$, $\rho_1 = \rho_2 = 1/200$); Bottom: W-NMF using Gibbs-OT. It is observed that the components of W-NMF with entropic regularization are smoother than those optimized with Gibbs-OT.

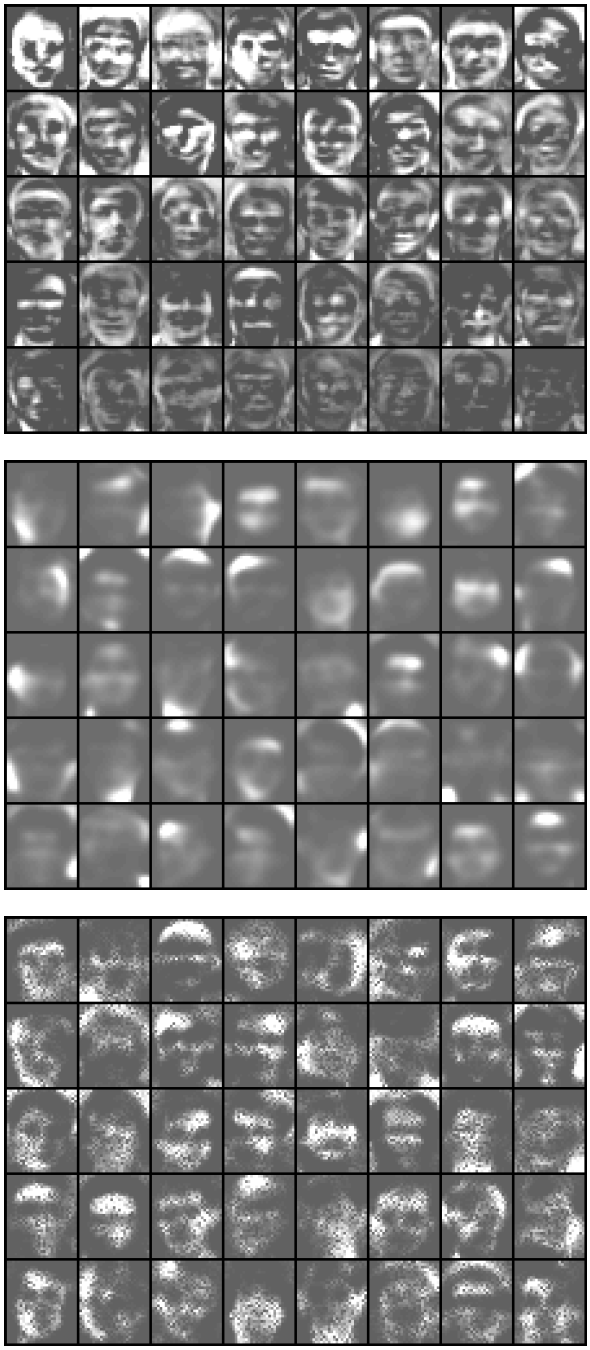


Figure 4.6. NMF components learned by different methods ($K = 40$) on the ORL face images. Top: regular NMF; Middle: W-NMF with entropic regularization ($\varepsilon = 1/100$, $\rho_1 = \rho_2 = 1/200$); Bottom: W-NMF using Gibbs-OT, in which the salt and pepper noises are observed due to the fact that Wasserstein distance is insensitive to the subpixel mass displacement [3].

Chapter 5
Determining Gains Acquired from Word Embedding: An Optimal Transport Application

5.1 Introduction

Word embeddings (a.k.a. word vectors) have been broadly adopted for document analysis [58, 68]. The embeddings can be trained from external large-scale corpus and then easily utilized for different data. To a certain degree, the knowledge mined from the corpus, possibly in very intricate ways, is coded in the vector space, the samples of which are easy to describe and ready for mathematical modeling. Despite the appeal, researchers will be interested in knowing how much gain an embedding can bring forth over the performance achievable by existing bag-of-words based approaches. Moreover, how can the gain be quantified? Such a preliminary evaluation will be carried out before building a sophisticated pipeline of analysis.

Almost every document analysis model used in practice is constructed assuming a certain basic representation—bag-of-words or word embeddings—for the sake of computational tractability. For example, after word embedding is done, high-level

The work presented in this chapter has been published in the form of a research paper: Jianbo Ye, Yanran Li, Zhaohui Wu, James Z. Wang, Wenjie Li, and Jia Li, "Determining Gains Acquired from Word Embeddings Quantitatively Using Discrete Distribution Clustering," *Proceedings of The Annual Meeting of the Association for Computational Linguistics (ACL)*, Vancouver, Canada, Vol 1, pp 1847–1856, July 2017.

models in the embedded space, such as entity representations, similarity measures, data manifolds, hierarchical structures, language models, and neural architectures, are designed for various tasks. In order to invent or enhance analysis tools, we want to understand precisely the pros and cons of the high-level models and the underlying representations. Because the model and the representation are tightly coupled in an analytical system, it is not easy to pinpoint where the gain or loss found in practice comes from. Should the gain be credited to the mechanism of the model or to the use of word embeddings? As our experiments demonstrate, introducing certain assumptions will make individual methods effective only if certain constraints are met. We will address this issue under an unsupervised learning framework.

Our proposed clustering paradigm has several advantages. Instead of packing the information of a document into a fixed-length vector for subsequent analysis, we treat a document more thoroughly as a distributional entity. In our approach, the distance between two empirical nonparametric measures (or discrete distributions) over the word embedding space is defined as the Wasserstein metric (a.k.a. the Earth Mover's Distance or EMD) [5, 56]. Comparing with a vector representation, an empirical distribution can represent with higher fidelity a cloud of points such as words in a document mapped to a certain space. In the extreme case, the empirical distribution can be set directly as the cloud of points. In contrast, a vector representation reduces data significantly, and its effectiveness relies on the assumption that the discarded information is irrelevant or nonessential to later analysis. This simplification itself can cause degradation in performance, obscuring the inherent power of the word embedding space.

Our approach is intuitive and robust. In addition to a high fidelity representation of the data, the Wasserstein distance takes into account the cross-term relationship between different words in a *principled* fashion. According to the definition, the distance between two documents A and B are the minimum cumulative cost that words from document A need to "travel" to match exactly the set of words for document B. Here, the travel cost of a path between two words is their (squared) Euclidean distance in the word embedding space. Therefore, how much benefit the Wasserstein distance brings also depends on how well the word embedding space captures the semantic difference between words.

While Wasserstein distance is well suited for document analysis, a major obstacle

of approaches based on this distance is the computational intensity, especially for the original D2-clustering method [19]. The main technical hurdle is to compute efficiently the Wasserstein barycenter, which is itself a discrete distribution, for a given set of discrete distributions. Thanks to the recent advances in the algorithms for solving Wasserstein barycenters [2, 8, 11, 21], one can now perform document clustering by directly treating them as empirical measures over a word embedding space. Although the computational cost is still higher than the usual vector-based clustering methods, we believe that the new clustering approach has reached a level of efficiency to justify its usage given how important it is to obtain high-quality clustering of unstructured text data. For instance, clustering is a crucial step performed ahead of cross-document co-reference resolution [69], document summarization, retrospective events detection, and opinion mining [70].

Our work has two main contributions. First, we create a basic tool of document clustering, which is easy to use and scalable. The new method leverages the latest numerical toolbox developed for optimal transport. It achieves state-of-the-art clustering performance across heterogeneous text data—an advantage over other methods in the literature. Second, the method enables us to quantitatively inspect how well a word-embedding model can fit the data and how much gain it can produce over the bag-of-words models.

5.2 Related Work

In the original D2-clustering framework proposed by [19], calculating Wasserstein barycenter involves solving a large-scale LP problem at each inner iteration, severely limiting the scalability and robustness of the framework. Such high magnitude of computations had prohibited it from deploying in many real-world applications until recently. To accelerate the computation of Wasserstein barycenter, and ultimately to improve D2-clustering, multiple numerical algorithmic efforts have been made in the recent few years [2, 8, 11, 21].

Although the effectiveness of Wasserstein distance has been well recognized in the computer vision and multimedia literature, the property of Wasserstein barycenter has not been well understood. To our knowledge, there still lacks systematic study of applying Wasserstein barycenter and D2-clustering in document analysis with word embeddings.

A closely related work by Kusner et al. [5] connects the Wasserstein distance to the word embeddings for comparing documents. Our work differs from theirs in the methodology. We directly pursue a scalable clustering setting rather than construct a nearest neighbor graph based on calculated distances, because the calculation of the Wasserstein distances of all pairs is too expensive to be practical. Kusner et al. [5] used a lower bound that was less costly to compute in order to prune unnecessary full distance calculation, but the scalability of this modified approach is still limited, an issue to be discussed in Section 5.4.3. On the other hand, our approach adopts the framework similar to the K-means which is of complexity $O(n)$ per iteration and usually converges within just tens of iterations. The computation of D2-clustering, though in its original form was magnitudes heavier than typical document clustering methods, can now be efficiently carried out with parallelization and proper implementations [8].

5.3 The Method

This section introduces the distance, the D2-clustering technique, the fast computation framework, and how they are used in the proposed document clustering method.

5.3.1 Wasserstein Distance

Suppose we represent each document d_k consisting m_k unique words by a discrete measure or a discrete distribution, where $k = 1, \ldots, N$ with N being the sample size:

$$d_k = \sum_{i=1}^{m_k} w_i^{(k)} \delta_{x_i^{(k)}} \ . \tag{5.1}$$

Here δ_x denotes the Dirac measure with support x, and $w_i^{(k)} \geq 0$ is the "importance weight" for the i-th word in the k-th document, with $\sum_{i=1}^{m_k} w_i^{(k)} = 1$. And $x_i^{(k)} \in \mathbb{R}^d$, called a support point, is the semantic embedding vector of the i-th word. The 2nd-order Wasserstein distance between two documents d_1 and d_2 (and likewise for

any document pairs) is defined by the following LP problem: $W^2(d_1, d_2) :=$

$$\begin{array}{ll} \min\limits_{\Pi} & \sum_{i,j} \pi_{i,j} \|x_i^{(1)} - x_j^{(2)}\|_2^2 \\ \text{s.t.} & \sum_{j=1}^{m_2} \pi_{i,j} = w_i, \forall i, \quad \sum_{i=1}^{m_1} \pi_{i,j} = w_j, \forall j \\ & \pi_{i,j} \geq 0, \forall i, j \ , \end{array} \tag{5.2}$$

where $\Pi = \{\pi_{i,j}\}$ is a $m_1 \times m_2$ coupling matrix, and let $\{C_{i,j} := \|x_i^{(1)} - x_j^{(2)}\|_2^2\}$ be transportation costs between words. Wasserstein distance is a true metric [1] for measures, and its best exact algorithm has a complexity of $O(m^3 \log m)$ [33], if $m_1 = m_2 = m$.

5.3.2 Discrete Distribution (D2-) Clustering

D2-clustering [19] iterates between the assignment step and centroids updating step in a similar way as the Lloyd's K-means. Suppose we are to find K clusters. The assignment step finds each member distribution its nearest mean from K candidates. The mean of each cluster is again a discrete distribution with m support points, denoted by c_i, $i = 1, \ldots, K$. Each mean is iteratively updated to minimize its total within cluster variation. We can write the D2-clustering problem as follows: given sample data $\{d_k\}_{k=1}^N$, support size of means m, and desired number of clusters K, D2-clustering solves

$$\min_{c_1, \ldots, c_K} \sum_{k=1}^N \min_{1 \leq i \leq K} W^2(d_k, c_i) \ , \tag{5.3}$$

where $c_1, \ldots, c_K$ are Wasserstein barycenters. At the core of solving the above formulation is an optimization method that searches the Wasserstein barycenters of varying partitions. Therefore, we concentrate on the following problem. For each cluster, we reorganize the index of member distributions from $1, \ldots, n$. The Wasserstein barycenter [2, 20] is by definition the solution of

$$\min_c \sum_{k=1}^n W^2(d_k, c) \ , \tag{5.4}$$

where $c = \sum_{i=1}^m w_i \delta_{x_i}$. The above Wasserstein barycenter formulation involves two levels of optimization: the outer level finding the minimizer of total variations, and the inner level solving Wasserstein distances. We remark that in D2-clustering, we need to solve multiple Wasserstein barycenters rather than a single one. This

constitutes the third level of optimization.

5.3.3 Modified Bregman ADMM for Wasserstein Barycenter

The recent modified Bregman alternating direction method of multiplier (B-ADMM) algorithm [8], motivated by the work by Wang and Banerjee [31], is a practical choice for computing Wasserstein barycenters. We briefly sketch their algorithmic procedure of this optimization method here for the sake of completeness. To solve for Wasserstein barycenter defined in Eq. (5.4), the key procedure of the modified Bregman ADMM involves iterative updates of four block of primal variables: the support points of c — $\{x_i\}_{i=1}^m$ (with transportation costs $\{C_{i,j}\}^{(k)}$ for $k = 1, \ldots, n$), the importance weights of c — $\{w_i\}_{i=1}^m$, and two sets of split matching variables — $\{\pi_{i,j}^{(k,1)}\}$ and $\{\pi_{i,j}^{(k,2)}\}$, for $k = 1, \ldots, n$, as well as Lagrangian variables $\{\lambda_{i,j}^{(k)}\}$ for $k = 1, \ldots, n$. In the end, both $\{\pi_{i,j}^{(k,1)}\}$ and $\{\pi_{i,j}^{(k,2)}\}$ converge to the matching weight in Eq. (5.2) with respect to $d(c, d_k)$. The iterative algorithm proceeds as follows until c converges or a maximum number of iterations are reached: given constant $\tau \geq 10$, $\rho \propto \dfrac{\sum_{i,j,k} C_{i,j}^{(k)}}{\sum_{k=1}^n m_k m}$ and round-off tolerance $\epsilon = 10^{-10}$, those variables are updated in the following order.

Update $\{x_i\}_{i=1}^m$ **and** $\{C_{i,j}^{(k)}\}$ in every τ iterations:

$$x_i := \frac{1}{n w_i} \sum_{k=1}^n \sum_{j=1}^{m_k} \pi_{i,j}^{(k,1)} x_j^{(k)}, \forall i, \tag{5.5}$$

$$C_{i,j}^{(k)} := \|x_i - x_j^{(k)}\|_2^2, \quad \forall i, j \text{ and } k. \tag{5.6}$$

Update $\{\pi_{i,j}^{(k,1)}\}$ **and** $\{\pi_{i,j}^{(k,2)}\}$. For each i, j and k,

$$\pi_{i,j}^{(k,2)} := \pi_{i,j}^{(k,2)} \exp\left(\frac{-C_{i,j}^{(k)} - \lambda_{i,j}^{(k)}}{\rho}\right) + \epsilon\,, \tag{5.7}$$

$$\pi_{i,j}^{(k,1)} := w_j^{(k)} \pi_{i,j}^{(k,2)} \Big/ \left(\sum_{l=1}^m \pi_{l,j}^{(k,2)}\right)\,, \tag{5.8}$$

$$\pi_{i,j}^{(k,1)} := \pi_{i,j}^{(k,1)} \exp\left(\lambda_{i,j}^{(k)}/\rho\right) + \epsilon\,. \tag{5.9}$$

Update $\{w_i\}_{i=1}^m$. For $i = 1, \ldots, m$,

$$w_i := \sum_{k=1}^n \frac{\sum_{j=1}^{m_k} \pi_{i,j}^{(k,1)}}{\sum_{i,j} \pi_{i,j}^{(k,1)}}\,, \tag{5.10}$$

$$w_i := w_i \Big/ \left(\sum_{i=1}^{m} w_i\right) . \tag{5.11}$$

Update $\{\pi_{i,j}^{(k,2)}\}$ **and** $\{\lambda_{i,j}^{(k)}\}$. For each i, j and k,

$$\pi_{i,j}^{(k,2)} := w_i \pi_{i,j}^{(k,1)} \Big/ \left(\sum_{l=1}^{m_k} \pi_{i,l}^{(k,1)}\right) , \tag{5.12}$$

$$\lambda_{i,j}^{(k)} := \lambda_{i,j}^{(k)} + \rho \left(\pi_{i,l}^{(k,1)} - \pi_{i,l}^{(k,2)}\right) . \tag{5.13}$$

Eq. (5.5)-(5.13) can all be vectorized as very efficient numerical routines. In a data parallel implementation, only Eq. (5.5) and Eq. (5.10) (involving $\sum_{k=1}^{n}$) needs to be synchronized. The software package detailed in [8] was used to generate relevant experiments. We make available our codes and pre-processed datasets for reproducing all experiments of our approach.

5.4 Experimental Results

5.4.1 Datasets and Evaluation Metrics

We prepare six datasets to conduct a set of experiments. Two short-text datasets are created as follows. (D1) BBCNews abstract: We concatenate the title and the first sentence of news posts from BBCNews dataset[1] to create an abstract version. (D2) Wiki events: Each cluster/class contains a set of news abstracts on the same story such as "2014 Crimean Crisis" crawled from Wikipedia current events following [52]; this dataset offers more challenges because it has more fine-grained classes and fewer documents (with shorter length) per class than the others. It also shows more realistic nature of applications such as news event clustering.

We also experiment with two long-text datasets and two domain-specific text datasets. (D3) Reuters-21578: We obtain the original Reuters-21578 text dataset and process as follows: remove documents with multiple categories, remove documents with empty body, remove duplicates, and select documents from the largest ten categories. Reuters dataset is a highly unbalanced dataset (the top category has more than 3,000 documents while the 10-th category has fewer than 100). This imbalance induces some extra randomness in comparing the results. (D4) 20News-

[1] BBCNews and BBCSport are downloaded from
http://mlg.ucd.ie/datasets/bbc.html

groups "bydate" version: We obtain the raw "bydate" version and process them as follows: remove headers and footers, remove URLs and Email addresses, delete documents with less than ten words. 20Newsgroups have roughly comparable sizes of categories. (D5) BBCSports. (D6) Ohsumed and Ohsumed-full: Documents are medical abstracts from the MeSH categories of the year 1991. Specifically, there are 23 cardiovascular diseases categories.

Evaluating clustering results is known to be nontrivial. We use the following three sets of quantitative metrics to assess the quality of clusters by knowing the ground truth categorical labels of documents: (i) Homogeneity, Completeness, and V-measure [53]; (ii) Adjusted Mutual Information (AMI) [6]; and (iii) Adjusted Rand Index (ARI) [59]. For sensitivity analysis, we use the homogeneity score [53] as a projection dimension of other metrics, creating a 2D plot to visualize the metrics of a method along different homogeneity levels. Generally speaking, more clusters leads to higher homogeneity by chance.

5.4.2 Methods in Comparison

We examine four categories of methods that assume a vector-space model for documents, and compare them to our D2-clustering framework. When needed, we use K-means++ to obtain clusters from dimension reduced vectors. To diminish the randomness brought by K-mean initialization, we ensemble the clustering results of 50 repeated runs [71], and report the metrics for the ensembled one. The largest possible vocabulary used, excluding word embedding based approaches, is composed of words appearing in at least two documents. On each dataset, we select the same set of Ks, the number of clusters, for all methods. Typically, Ks are chosen around the number of ground truth categories in logarithmic scale.

We prepare two versions of the TF-IDF vectors as the unigram model. The ensembled K-means methods are used to obtain clusters. (1) *TF-IDF* vector [72]. (2) *TF-IDF-N* vector is found by choosing the most frequent N words in a corpus, where $N \in \{500, 1000, 1500, 2000\}$. The difference between the two methods highlights the sensitivity issue brought by the size of chosen vocabulary.

We select three dimensionality reduction methods. After the dimensionality is reduced, ensembled K-means methods are used to obtain clusters. We remark that most manifold learning approaches, including those we've experimented, are

transductive, scaling quadratically w.r.t. the sample size. This makes those approaches difficult to be applied in online or large-scale clustering settings. (3) *Spectral Clustering (Laplacian).* We apply the Laplacian Eigenmaps [73] with varying numbers of components to reduce the dimension of Tfidf vectors. Their cosine affinities are constructed based on nearest neighbors. (4) *Latent Semantic Indexing (LSI)* [74]. We compute SVD of the Tfidf matrix, and choose varying numbers of components to form the dimension reduced vectors. (5) *Locality Preserving Projection (LPP)* [75, 76]. LPP is a popular document indexing method which produces low-dimensional representations. We follow the suggested setup in Cai *et al.* [76] and use the cosine affinity matrix.[2] We note that extra hyper-parameters are chosen from a pre-selected set empirically achieving the best performances.

We select two topic models. topic modeling techniques are originally developed to characterize documents with multiple topics, rather than cluster them into disjoint groups. Nevertheless, by assigning each document to its most significant topic, a clustering result can be obtained. We highlight that mixture-of-topics assumption that commonly utilized in topic modeling makes many of their approaches less sensitive to explore the homogeneity of clusters when increasing topics are estimated. (6) *Non-negative Matrix Factorization (NMF)* [77, 78]. We compute NMF of the Tfidf matrix by choosing K components, where K is the desired number of clusters. Documents (by row) are assigned to their largest column respectively in the factorization matrix to form clusters. (7) *Latent Dirichlet Allocation (LDA)* [61, 79]. Similar to NMF, we solve a LDA model of the word counting matrix by setting K topics. Because there are many hyper-parameters in a LDA model, our chosen set of hyper-parameters are set to achieve the low perplexity empirically. Adapting LDA to clustering, we naively group documents based on their most likely topics. Empirically, we find it works better than K-means of topic proportion vectors of documents.

Three methods based on word embeddings: we use four pre-trained word embeddings in our experiment to cross validate the effects of different pre-trained word embeddings. Three of them are trained on general large corpora, such as news articles and wikipedia pages. They are 300-dimensional SkipGram [58] using negative sampling trained on GoogleNews, 300-dimensional Glove [57] trained on Wikipedia corpus, standard 400-dimensional SkipGram trained on a 2010 Wikipedia

[2] http://www.cad.zju.edu.cn/home/dengcai/Data/DimensionReduction.html

dump[3] by our own (with window size of 10 and minimum count of 10). The first two can be downloaded publicly.[4] When compared with other non-embedding methods, best results of the three are reported.

We also use SkipGram to train domain-specific word embeddings using Ohsumed dataset (with window size of 20 and minimum count of 2), which is the fourth model. (8) *Average of word vectors (AvgDoc)* computes the average of the embeddings of distinctive words in a document. In practice, we find it outperforms one using weighted schema like Tfidf or Tf, especially for long texts. (9) *Paragraph Vectors (PV)* [80]. Two unsupervised methods, *i.e.* PVDM and PVDBOW, are proposed in Le and Mikolov [80], in which pre-trained word vectors can be fine-tuned to obtain embeddings for documents. We have experimented with both methods PVDM and PVDBOW, and find PVDM performs significantly worse than PVDBOW on all datasets, thus only the results of PVDBOW are reported.

5.4.3 Runtime

We report the runtime for our approach on two largest datasets. The experiments regarding other smaller datasets all terminate within minutes in a single machine, which we omit due to space limitation. Like K-means, the runtime by our approach depends on the number of actual iterations before a termination criterion is met. In the Newsgroups dataset, with $m = 100$ and $K = 45$, the time per iteration is 121 seconds on 48 processors. In Reuters dataset, with $m = 100$ and $K = 20$, the time per iteration is 190 seconds on 24 processors. Each run terminates in around tens of iterations typically, upon which the percentage of label changes is less than 0.1%.

Our approach adopts the Elkan's algorithm [50] pruning unnecessary computations of Wasserstein distance in assignment steps of K-means. For the Newsgroups data (with $m = 100$ and $K = 45$), our approach terminates in 36 iterations, and totally computes $12,162,717$ ($\approx 3.5\% \times 18612^2$) distance pairs in assignment steps, saving 60% ($\approx 1 - \frac{12,162,717}{36\times45\times18612}$) distance pairs to calculate in the standard D2-clustering. In comparison, the clustering approaches based on K-nearest neighbor (KNN) graph with the prefetch-and-prune method of [5] needs substantially more

[3] http://www.psych.ualberta.ca/~westburylab/downloads/westburylab.wikicorp.download.html

[4] https://code.google.com/p/word2vec/ http://nlp.stanford.edu/projects/glove/

pairs to compute Wasserstein distance, meanwhile the speed-ups also suffer from the curse of dimensionality. Their detailed statistics are reported in Table 5.1. Based on the results, our approach is much more practical as a basic document clustering tool.

Method		EMD counts (%)
Our approach	$d = 400$, $K = 10$	**2.0**
Our approach	$d = 400$, $K = 40$	**3.5**
KNN	$d = 400$, $K = 1$	73.9
KNN	$d = 100$, $K = 1$	53.0
KNN	$d = 50$, $K = 1$	23.4

Table 5.1. Percentage of total 18612^2 Wasserstein distance pairs needed to compute on the full Newsgroup dataset. The KNN graph based on 1st order Wasserstein distance is computed from the prefetch-and-prune approach according to [5].

5.4.4 Results

We now summarize our numerical results.

Dataset	size	class	length	est. #voc.
BBCNews abstr.	2,225	5	26	7,452
Wiki events	1,983	54	22	5,313
Reuters	7,316	10	141	27,792
Newgroups	18,612	20	245	55,970
BBCSports	737	5	345	13,105
Ohsumed	4,340	23	-	-
Ohsumed-full*	34,386	23	184	43,895

Table 5.2. Description of corpus data that have been used in our experiments. *Ohsumed-full dataset is used for pre-training word embeddings only. Ohsumed is a downsampled evaluation set resulting from removing posts from Ohsumed-full that belong to multiple categories.

Regular text datasets. The first four datasets in Table 5.2 cover quite general and broad topics. We consider them to be regular and representative datasets encountered more frequently in applications. We report the clustering performances of the ten methods in Fig. 5.1, where three different metrics are plotted against the clustering homogeneity. The higher result at the same level of homogeneity is

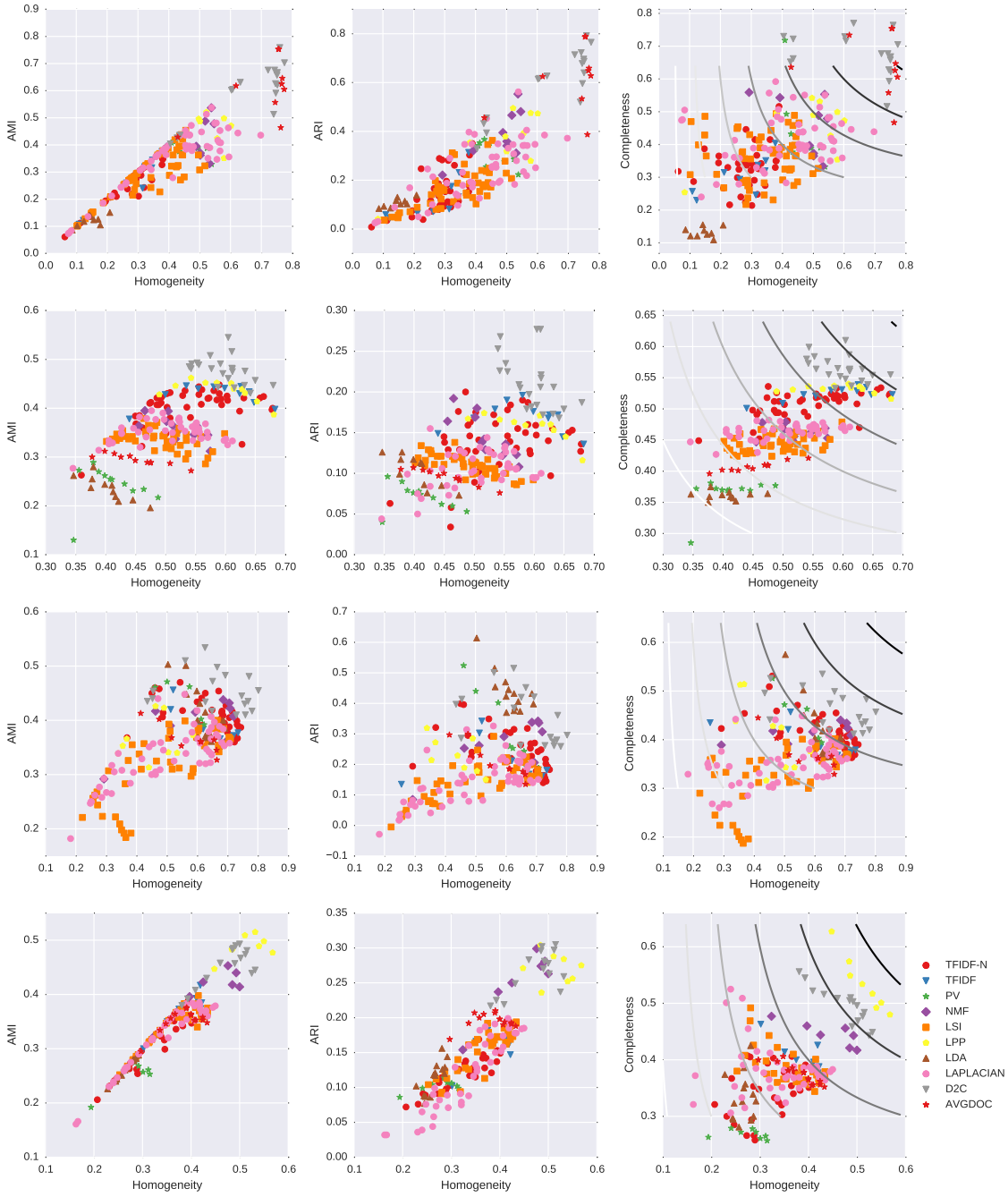


Figure 5.1. The quantitative cluster metrics used for performance evaluation of "BBC title and abstract", "Wiki events", "Reuters", and "Newsgroups" (row-wise, from top to down). Y-axis corresponds to AMI, ARI, and Completeness, respective (column-wise, from left to right). X-axis corresponds to Homogeneity for sensitivity analysis.

better, and the ability to achieve higher homogeneity is also welcomed. *Clearly, D2-clustering is the only method that shows robustly superior performances among all ten methods.* Specifically, it ranks first in three datasets, and second in the other one. In comparison, LDA performs competitively on the "Reuters" dataset, but is substantially unsuccessful on others. Meanwhile, LPP performs competitively on the "Wiki events" and "Newsgroups" datasets, but it underperforms on the other two. Laplacian, LSI, and Tfidf-N can achieve comparably performance if their reduced dimensions are fine tuned, which unfortunately is unrealistic in practice. NMF is a simple and effective method which always gives stable, though subpar, performance.

Short texts vs. long texts. D2-clustering performs much more impressively on short texts ("BBC abstract" and "Wiki events") than it does on long texts ("Reuters" and "Newsgroups"). This outcome is somewhat expected, because the bag-of-words method suffers from high sparsity for short texts, and word-embedding based methods in theory should have an edge here. As shown in Fig. 5.1, D2-clustering has indeed outperformed other non-embedding approaches by a large margin on short texts (improved by about 40% and 20% respectively). Nevertheless, we find lifting from word embedding to document clustering is not without a cost. Neither AvgDoc nor PV can perform as competitively as D2-clustering performs on both.

	regular dataset				domain-specific dataset		
	BBCNews abstract	Wik events	Reuters	Newsgroups	BBCSport	Ohsumed	Avg.
Tfidf-N	0.389	0.448	0.470	0.388	**0.883**	0.210	0.465
Tfidf	0.376	0.446	0.456	0.417	0.799	0.235	0.455
Laplacian	0.538	0.395	0.448	0.385	0.855	0.223	0.474
LSI	0.454	0.379	0.400	0.398	0.840	0.222	0.448
LPP	0.521	0.462	0.426	**0.515**	0.859	**0.284**	0.511
NMF	0.537	0.395	0.438	0.453	0.809	0.226	0.476
LDA	0.151	0.280	0.503	0.288	0.616	0.132	0.328
AvgDoc	**0.753**	0.312	0.413	0.376	0.504	0.172	0.422
PV	0.428	0.289	0.471	0.275	0.553	0.233	0.375
D2C	**0.759**	**0.545**	**0.534**	0.493	0.812	0.260	**0.567**

Table 5.3. Best AMIs [6] of compared methods on different datasets and their averaging. The best results are marked in bold font for each dataset, the 2nd and 3rd are marked by blue and magenta colors respectively.

Domain-specific text datasets. We are also interested in how word embedding can help group domain-specific texts into clusters. In particular, does the semantic

knowledge "embedded" in words provides enough clues to discriminate fine-grained concepts? We report the best AMI achieved by each method in Table 5.3. Our preliminary result indicates state-of-the-art word embeddings do not provide enough gain here to exceed the performance of existing methodologies. On the unchallenging one, the "BBCSport" dataset, basic bag-of-words approaches (Tfidf and Tfidf-N) already suffice to discriminate different sport categories; and on the difficult one, the "Ohsumed" dataset, D2-clustering only slightly improves over Tfidf and others, ranking behind LPP. Meanwhile, we feel the overall quality of clustering "Ohsumed" texts is quite far from useful in practice, no matter which method to use. More discussions will be provided next.

5.4.5 Sensitivity to Word Embeddings.

We validate the robustness of D2-clustering with different word embedding models, and we also show all their results in Fig. 5.2. As we mentioned, the effectiveness of Wasserstein document clustering depends on how relevant the utilized word embeddings are with the tasks. In those general document clustering tasks, however, word embedding models trained on general corpus perform robustly well with acceptably small variations. This outcome reveals our framework as generally effective and not dependent on a specific word embedding model. In addition, we also conduct experiments with word embeddings with smaller dimensions, at 50 and 100. Their results are not as good as those we have reported (therefore detailed numbers are not included due to space limitation).

Inadequate embeddings may not be disastrous. In addition to our standard running set, we also used D2-clustering with purely random word embeddings, meaning each word vector is independently sampled from spherical Gaussian at 300 dimension, to see how deficient it can be. Experimental results show that random word embeddings degrade the performance of D2-clustering, but it still performs much better than purely random clustering, and is even consistently better than LDA. Its performances across different datasets is highly correlated with the bag-of-words (Tfidf and Tfidf-N). By comparing a pre-trained word embedding model to a randomly generated one, we find that the extra gain is significant ($> 10\%$) in clustering four of the six datasets. Their detailed statistics are in Table 5.4 and Fig. 5.3.

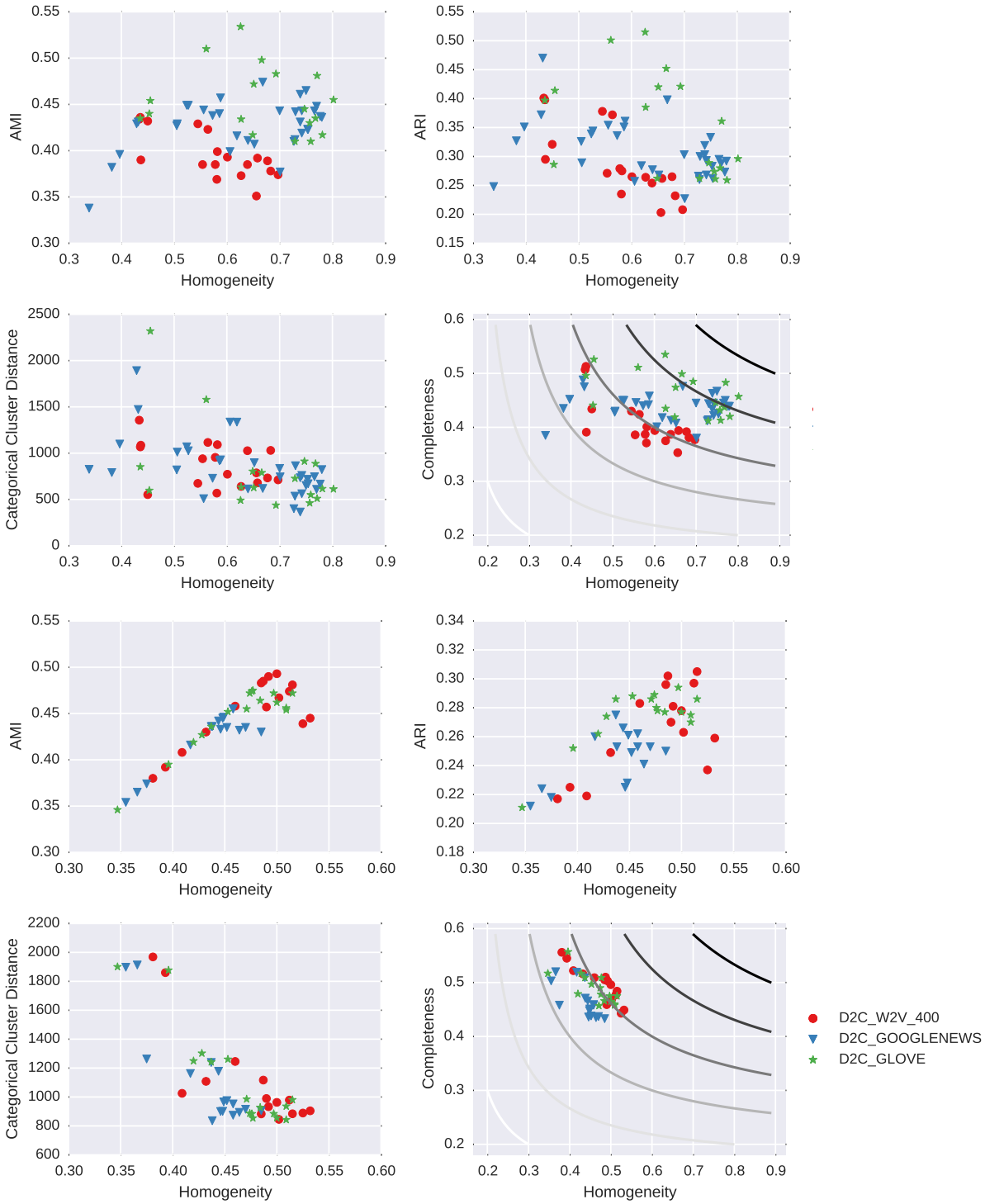


Figure 5.2. Sensitivity analysis: the clustering performances of D2C under different word embeddings. Upper: Reuters, Lower: Newsgroups. An extra evaluation index (CCD [4]) is also used.

	ARI	AMI	V-measure
BBCNews	.146	.187	.190
abstract	$.792_{+442\%}$	$.759_{+306\%}$	$.762_{+301\%}$
Wiki events	.194	.369	.463
	$.277_{+43\%}$	$.545_{+48\%}$	$.611_{+32\%}$
Reuters	.498	.524	.588
	$.515_{+3\%}$	$.534_{+2\%}$	$.594_{+1\%}$
Newsgroups	.194	.358	.390
	$.305_{+57\%}$	$.493_{+38\%}$	$.499_{+28\%}$
BBCSport	.755	.740	.760
	$.801_{+6\%}$	$.812_{+10\%}$	$.817_{+8\%}$
Ohsumed	.080	.204	.292
	$.116_{+45\%}$	$.260_{+27\%}$	$.349_{+20\%}$

Table 5.4. Comparison between *random* word embeddings (upper row) and meaningful *pre-trained* word embeddings (lower row), based on their best ARI, AMI, and V-measures. The improvements by percentiles are also shown in the subscripts.

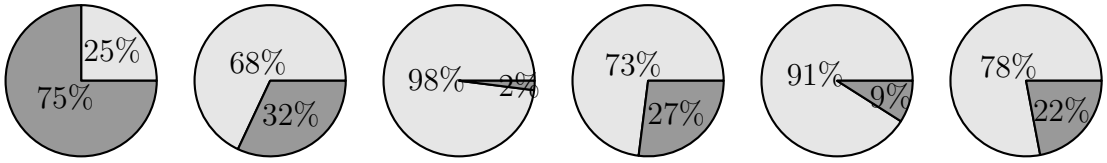


Figure 5.3. Pie charts of clustering gains in AMI calculated from our framework. Light region is by bag-of-words, and dark region is by pre-trained word embeddings. Six datasets (from left to right): BBCNews abstract, Wiki events, Reuters, Newsgroups, BBCSport, and Ohsumed.

5.5 Discussions

Performance advantage. There has been one immediate observation from these studies, D2-clustering always outperforms two of its degenerated cases, namely Tf-idf and AvgDoc, and three other popular methods: LDA, NMF, and PV, on all tasks. Therefore, for document clustering, users can expect to gain performance improvements by using our approach.

Clustering sensitivity. From the four 2D plots in Fig. 5.1, we notice that the results of Laplacian, LSI and Tfidf-N are rather sensitive to their extra hyper-parameters. Once the vocabulary set, weight scheme and embeddings of words are

fixed, our framework involves only two additional hyper-parameters: the number of intended clusters, K, and the selected support size of centroid distributions, m. We have chosen more than one m in all related experiments ($m = \{64, 100\}$ for long documents, and $m = \{10, 20\}$ for short documents). Our empirical experiments show that the effect of m on different metrics is less sensitive than the change of K. Results at different K are plotted for each method (Fig. 5.1). The gray dots denote results of multiple runs of D2-clustering. They are always contracted around the top-right region of the whole population, revealing the predictive and robustly supreme performance.

When bag-of-words suffices. Among the results of "BBCSport" dataset, Tfidf-N shows that by restricting the vocabulary set into a smaller one (which may be more relevant to the interest of tasks), it already can achieve highest clustering AMI without any other techniques. Other unsupervised regularization over data is likely unnecessary, or even degrades the performance slightly.

Toward better word embeddings. Our experiments on the Ohsumed dataset have been limited. The result shows that it could be highly desirable to incorporate certain domain knowledge to derive more effective vector embeddings of words and phrases to encode their domain-specific knowledge, such as jargons that have knowledge dependencies and hierarchies in educational data mining, and signal words that capture multi-dimensional aspects of emotions in sentiment analysis.

Finally, we report the best AMIs of all methods on all datasets in Table 5.3. By looking at each method and the average of best AMIs over six datasets, we find our proposed clustering framework often performs competitively and robustly, which is the only method reaching more than 90% of the best AMI on each dataset. Furthermore, this observation holds for varying lengths of documents and varying difficulty levels of clustering tasks.

This chapter introduces a nonparametric clustering framework for document analysis. Its computational tractability, robustness and supreme performance, as a fundamental tool, are empirically validated. Its ease of use enables data scientists to apply it for the pre-screening purpose of examining word embeddings in a specific task. Finally, the gains acquired from word embeddings are quantitatively measured from a nonparametric unsupervised perspective.

It would also be interesting to investigate several possible extensions to the current clustering work. One direction is to learn a proper ground distance for

word embeddings such that the final document clustering performance can be improved with labeled data. The work by [81,82] have partly touched this goal with an emphasis on document proximities. A more appealing direction is to develop problem-driven methods to represent a document as a distributional entity, taking into consideration of phrases, sentence structures, and syntactical characteristics. We believe the framework of Wasserstein distance and D2-clustering creates room for further investigation on complex structures and knowledge carried by documents.

Chapter 6
Improving the Quality of Crowdsourced Affective Data: A Probabilistic Modeling Application

6.1 Introduction

Humans' sensitivity to affective stimuli intrinsically varies from one person to another. Differences in gender, age, society, culture, personality, social status, and personal experience can contribute to its high variability between people. Further, inconsistencies may also exist for the same individual across environmental contexts and current mood or affective state. The causal effects and factors for such affective experiences have been extensively investigated, as evident in the literature on psychological and human studies, where controlled experiments are commonly conducted within a small group of human subjects — to ensure the reliability of collected data. To complement the shortcomings of those controlled experiments, ecological psychology aims to understand how objects and things in our surrounding environments effect human behaviors and affective experiences, in which *real-world* studies are favored over those within artificial laboratory environments. The key

The work presented in this chapter has been published in the form of a research paper: Jianbo Ye, Jia Li, Michelle G. Newman, Reginald B. Adams, Jr. and James Z. Wang, "Probabilistic Multigraph Modeling for Improving the Quality of Crowdsourced Affective Data," *IEEE Transactions on Affective Computing (TAC)*, 2017, 15 pages. https://doi.org/10.1109/TAFFC.2017.2678472

ingredient of those ecological approaches is the availability of large-scale data collected from human subjects, remedying the high complexity and heterogeneity that the real-world has to offer. With the growing attention on affective computing, multiple data-driven approaches have been developed to understand what particular environmental factors drive the feelings of humans [83, 84], and how those effects differ among various sociological structures and between human groups.

One crucial hurdle for those affective computing approaches is the lack of full-spectrum annotated stimuli data at a large scale. To address this bottleneck, crowdsourcing-based approaches are highly helpful for collecting uncontrolled human data from anonymous participants. In a recent study reported in [85], anonymous subjects from the Internet were recruited to annotate a set of visual stimuli (images): at each time point, after being presented with an image stimulus, participants were asked to assess their personal psychological experiences using ordinal scales for each of the affective dimensions: valence, arousal, dominance and likeness (which means the degree of appreciation in our context). This study also collected demographics data to analyze individual difference predictors of affective responses. Because labeling a large number of visual stimuli can become tedious, even with crowdsourcing, each image stimulus was examined by only a few subjects. This study allowed tens of thousands of images to obtain at least one label from a participant, which created a large data set for environmental psychology and automated emotion analysis of images.

One interesting question to investigate, however, is *whether the affective labels provided by subjects are reliable*. A related question is how to separate spammers from reliable subjects, or at least to narrow the scope of data to a highly reliable subgroup. Here, spammers are defined as those participants who provide answers without serious consideration of the presented questions. No answer from a statistical perspective is known yet for crowdsourced affective data.

A great difficulty in analyzing affective data is caused by the absence of ground truth in the first place, that is, there is no *correct* answer for evoked emotion. It is generally accepted that even the most reliable subjects can naturally have varied emotions. Indeed, with variability among human responses anticipated, psychological studies often care about questions such as where humans are emotionally consistent and where they are not, and which subgroups of humans are more consistent than another. Given a population, many, if not the vast majority

of stimuli may not have a consensus emotion at all. Majority voting or (weighted) averaging to force an "objective truth" of the emotional response or probably for the sake of convenience, as is routinely done in affective computing so that classification on a single quantity can be carried out, is a crude treatment bound to erase or disregard information essential for many interesting psychological studies, *e.g.*, to discover connections between varied affective responses and varied demographics.

The involvement of spammers as participating subjects introduces an extra source of variation to the emotional responses, which unfortunately is tangled with the "appropriate" variation. If responses associated with an image stimulus contain answers by spammers, the inter-annotator variation for the specific question could be as large as the variation across different questions, reducing the robustness of any analysis. An example is shown in Fig. 6.1. Most annotators labeling this image are deemed unreliable, and two of them are highly susceptible as spammers according to our model. Investigators may be recommended to eliminate this image or acquire more reliable labels for its use. Yet, one should not be swayed by this example into the practice of discarding images that solicited responses of a large range. Certain images are controversial in nature and will stimulate quite different emotions to different viewers. Our system acquired the reliability scores shown in Fig. 6.1 by examining the entire data set; the data on this image alone would not be conclusive, in fact, far from so.

Facing the intertwined "appropriate" and "inappropriate" variations in the subjects as well as the variations in the images, we are motivated to unravel the sources of uncertainties by taking a global approach. The judgment on the reliability of a subject cannot be a per-image decision, and has to leverage the whole data. Our model was constructed to integrate these uncertainties, attempting to discern them with the help of big data. In addition, due to the lack of ground truth labels, we model the relational data that code whether two subjects' emotion responses on an image agree, bypassing the thorny questions of what the true labels are and if they exist at all.

For the sake of automated emotion analysis of images, one also needs to narrow the scope to parts of data, each of which have sufficient number of qualified labels. Our work computes image confidences, which can support off-line data filtering or guide on-line budgeted crowdsourcing practices.

In summary, systematic analysis of crowdsourced affective data is of great

Annotator ID	Valence	Reliability
3474	5.1/8	0.08
2500	0.0/8	0.56
3475	0.0/8	0.34
2540	8.0/8	0.04

Image Confidence: 75% ($\leq$ 90%)

Figure 6.1. An example illustrating one may need to acquire more reliable labels, ensuring the image confidence is more than 0.9.

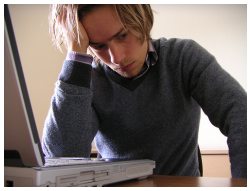

raw avg.: 4.06 out of 8
$\rightarrow$ new: 2.51 out of 8


4.1 $\rightarrow$ 2.94

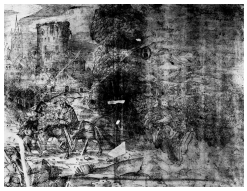

4.78 $\rightarrow$ 3.33

4.25 $\rightarrow$ 1.9


4.06 $\rightarrow$ 3.03


4.05 $\rightarrow$ 2.87


4.7 $\rightarrow$ 2.06


5.08 $\rightarrow$ 3.94


5.02 $\rightarrow$ 3.58


5.7 $\rightarrow$ 3.87

5.6 $\rightarrow$ 3

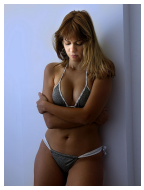

5.17 $\rightarrow$ 3.19


5.32 $\rightarrow$ 2.98


5.5 $\rightarrow$ 4.26


4.53 $\rightarrow$ 3


5.24 $\rightarrow$ 3.93


5.38 $\rightarrow$ 3.76


4.54 $\rightarrow$ 2.75

Figure 6.2. Images shown are considered of lower valence than their average valence ratings (*i.e.*, evoking a higher degree of negative emotions) after processing the data set using our proposed method. Our method eliminates the contamination introduced by spammers. The range of valence ratings is between 0 and 8.

2.63 → 3.77 2.8 → 4.14 3.0 → 4.7 4.4 → 6.21 4.7 → 6.26

Figure 6.3. Images shown are considered of higher valence than their average valence ratings (*i.e.*, evoking a higher degree of positive emotions) after processing the data set using our proposed method. Our method again eliminates the contamination introduced by spammers. The range of valence ratings is between 0 and 8.

importance to human subject studies and affective computing, while remains an open question. To substantially address the aforementioned challenges and expand the evidential space for psychological studies, we propose a probabilistic approach, called **Gated Latent Beta Allocation (GLBA)**. This method computes maximum a posteriori probability (MAP) estimates of each subject's reliability and regularity based on a variational expectation-maximization (EM) framework. With this method, investigators running affective human subject studies can substantially reduce or eliminate the contamination caused by spammers, hence improve the quality and usefulness of collected data (Fig. 6.2).

6.1.1 Related Work

Estimating the reliability of subjects is necessary in crowdsourcing-based data collection because the incentives of participants and the interest of researchers diverge. There were two levels of assumptions explored for the crowdsourced data, which we name as the first-order assumption (A1) and the second-order assumption (A2). Let a task be the provision of emotion responses for one image. Consider a task or test conducted by a number of participants. Their responses within this task form a subgroup of data.

A1 There exists a true label of practical interest for each task. The dependencies between collected labels are mediated by this unobserved true label, of which noisy labels are otherwise conditionally independent.

A2 The uncertainty model for a subgroup of data does not depend on its actual specified task. The performance of a participant is consistent across subgroups of data subject to a single fixed effect.

Existing approaches that model the complexities of tasks or reliability of participants often require one or both of these two assumptions. Under the umbrella of assumption A1, most probabilistic approaches using the observer models [86–89] focus on estimating the ground truth from multiple noisy labels. For example, the modeling of one reliability parameter per subject is an established practice for estimating the ground truth label [89]. For the case of categorical labels, modeling of one free parameter per class per subject is a more general approach [86, 90]. Our approach does not model the ground truth of labels, hence it is not viable to compare our approach with other methods in this regard. Instead, we sidestep this issue to tackle whether the labels from one subject can agree with labels from another on a single task. Agreement is judged subject to a preselected criterion. Such treatment may be more realistic as a means to process sparse ordinal labels for each task.

Assumption A2 is also widely exploited among methods, often conditioned on A1. It assumes that all of the tasks have the same level of difficulty [91, 92]. Modeling one difficulty parameter per task has been explored in [93] for categorical labels. However, in our approach, task difficulty is modeled as a random effect without subscribing a task-specific parameter. Wisely choosing the modeling complexity and assumptions should be based on availability and purity of data. As suggested in [94], more complexity in a model could challenge the statistical estimation subject to the constraint of real data. Choices with respect to our model attempted to properly analyze the affective data we obtained.

If the mutual agreement rate between two participants does not depend on the actual specified task (*i.e.*, when A2 holds), we can essentially convert the resulting problem to a graph mining problem, where subjects are vertices, agreements are edges, and the proximity between subjects is modeled by how likely they agree with each other in a general sense. Probabilistic models for such relational data can be traced back to early stochastic blockmodels [95, 96], latent space model [97], and their later extensions with mixed membership [98, 99] and nonparametric Bayes [100]. We adopt the idea of mixed memberships wherein two particular modes of memberships are modeled for each subject, one being the reliable mode and the other the random mode. For the random mode, the behavior is assumed to be shared across different subjects, whereas the regular behaviors of subjects in the reliable mode are assumed to be different. Therefore, we can extend this framework

from graph to multigraph in the interest of crowdsourced data analysis. Specifically, data are collected as subgroups, each of which is composed of a small agreement graphs for a single task, such that the covariate within a subgroup is modeled. Our approach does not rely on A2. Instead, it models the random effects added to subjects' performance in each task via the multigraph approach. Assumption A1 and A2 implies a bipartite graph structure between tasks and subjects. In contrast, our approach starts from the multigraph structure among subjects that is coordinated by tasks. Finding the proper and flexible structure that data possess is crucial for modeling [101].

6.1.2 Our Contributions

To our knowledge, this is the first attempt to connect probabilistic observer models with probabilistic graphs, and to explore modeling at this complexity from the joint perspective. We summarize our contributions as follows:

- We developed a probabilistic multigraph model to analyze crowdsourced data and its approximate variational EM algorithm for estimation. The new method, accepting the intrinsic variation in subjective responses, does not assume the existence of ground truth labels, in stark contrast to previous work having devoted much effort to obtain objective true labels.

- Our method exploits the relational data in the construction and application of the statistical model. Specifically, instead of the direct labels, the pair-wise status of agreement between labels given by different subjects is used. As a result, the multigraph agreement model is naturally applicable to more flexible types of responses, easily going beyond binary and categorical labels. Our work serves as a proof of concept for this new relational perspective.

- Our experiments have validated the effectiveness of our approach on real-world affective data. Because our experimental setup was of a larger scale and more challenging than settings addressed by existing methods, we believe our method can fill some gaps for demands in the practical world, for instance, when gold standards are not available.

6.2 The Method

In this section, we describe our proposed method. Let us present the mathematical notations first. A symbol with subscript omitted always indicates an array, *e.g.*, $x = (\ldots, x_i, \ldots)$. The arithmetic operations perform over arrays in the element-wise manner, *e.g.*, $x + y = (\ldots, x_i + y_i, \ldots)$. Random variables are denoted as capital English letters. The tilde sign indicates the value of parameters in the last iteration of EM, *e.g.*, $\tilde{\theta}$. Given a function f_θ, we denote $f_{\tilde{\theta}}$ by $\tilde{f}_\theta$ or simply $\tilde{f}$, if the parameter $\tilde{\theta}$ is implied. Additional notations, as summarized in Table 6.1, will be explained in more details later.

Symbols	Descriptions
O_i	subject i
τ_i	rate of subject reliability
α_i, β_i	shape of subject regularity
γ	rate of agreement by chance
Θ	union of parameters
$T_j^{(k)}$	whether O_j reliably response
$J_i^{(k)}$	rate of O_i agreeing with other reliable responses
$I_{i,j}^{(k)}$	whether O_i agrees with the responses from O_j
$\omega_i^{(k)}(\cdot)$	cumulative degree of responses agreed by O_i
$\psi_i^{(k)}(\cdot)$	cumulative degree of responses
$r_j^{(k)}(\cdot)$	a ratio amplifies or discounts the reliability of O_j
$\tilde{\tau}_i^{(k)}$	sufficient statistics of posterior $T_i^{(k)}$, given $\tilde{\Theta}$
$\tilde{\alpha}_i^{(k)}, \tilde{\beta}_i^{(k)}$	sufficient statistics of posterior $J_i^{(k)}$, given $\tilde{\Theta}$

Table 6.1. Symbols and descriptions of parameters, random variables, and statistics.

6.2.1 Agreement Multigraph

We represent the data as a directed multigraph, which does not assume a particular type of crowdsourced response. Suppose we have prepared m questions in the study, the answers can be binary, categorical, ordinal, and multidimensional. Given a subject pair (i, j) who are asked to look at the k-th question, one designs an agreement protocol that determines whether the answer from subject i agrees with

that from subject j. If subject i's agrees with subject j's on task k, then we set $I_{i,j}^{(k)} = 1$. Otherwise, $I_{i,j}^{(k)} = 0$.

In our case, we are given ordinal data from multiple channels, we define $I_{i,j}^{(k)} = 1$ if (sum of) the percentile difference between two answers $a_i, a_j \in \{1, \ldots, A\}$ satisfies

$$\frac{1}{2}\left|P\left[a_i^{(k)}\right] - P\left[a_j^{(k)}\right]\right| + \frac{1}{2}\left|P\left[a_i^{(k)}+1\right] - P\left[a_j^{(k)}+1\right]\right| \leq \delta. \tag{6.1}$$

The percentile $P[\cdot]$ is calculated from the whole pool of answers for each discrete value, and $\delta = 0.2$. In the above equation, we measure the percentile difference between a_i and a_j as well as that between $a_i + 1$ and $a_j + 1$ in order to reduce the effect of imposing discrete values on the answers that are by nature continuous. If the condition does not hold, they disagree and $I_{i,j}^{(k)} = 0$. Here we assume that if two scores for the same image are within a 20% percentile interval, they are considered to reach an agreement. Compared with setting a threshold on their absolute difference, such rule adapts to the non-uniformity of score distribution. Two subjects can agree with each other by chance or they indeed experience similar emotions in response to the same visual stimulus.

While the choice of the percentile threshold δ is inevitably subjective, the selection in our experiments was guided by the desire to trade-off the preservation of the original continuous scale of the scores (favoring small values) and a sufficient level of error tolerance (favoring large values). This threshold controls the sparsity level of the multi-graph, and influences the marginal distribution of estimated parameters. Alternatively, one may assess different values of the threshold and make a selection based on some other criteria of preference (if exist) applied to the final results.

6.2.2 Gated Latent Beta Allocation

This subsection describes the basic probabilistic graphical model we used to jointly model subject reliability, which is independent from the supplied questions, and regularity. We refrain from carrying out a full Bayesian inference because it is impractical to end users. Instead, we use the mode(s) of the posterior as point estimates.

We assume each subject i has a reliability parameter $\tau_i \in [0, 1]$ and regularity parameters $\alpha_i,\ \beta_i > 0$ characterizing his or her agreement behavior with the

population, for $i = 1, \ldots, m$. We also use parameter γ for the rate of agreement between subjects out of pure chance. Let $\Theta = (\{\tau_i, \alpha_i, \beta_i\}_{i=1}^m, \gamma)$ be the set of parameters. Let Ω_k be the a random sub-sample from subjects $\{1, \ldots, m\}$ who labeled the stimulus k, where $k = 1, \ldots, n$. We also assume sets Ω_k's are created independently from each other. For each image k, every subject pair from Ω_k^2, *i.e.*, (i, j) with $i \neq j$, has a binary indicator $I_{i,j}^{(k)} \in \{0, 1\}$ coding whether their opinions agree on the respective stimulus. We assume $I_{i,j}^{(k)}$ are generated from the following probabilistic process with two latent variables. The first latent variable $T_j^{(k)}$ indicates whether subject O_j is reliable or not. Given that it is binary, a natural choice of model is the Bernoulli distribution. The second latent variable $J_i^{(k)}$, lying between 0 and 1, measures the extent subject O_i agrees with the other reliable responses. We use Beta distribution parameterized by α_i and β_i to model $J_i^{(k)}$ because it is a widely used parametric distribution for quantities on interval $[0, 1]$ and the shape of the distribution is relatively flexible. In a nutshell, $T_j^{(k)}$ is a latent switch (aka, gate) that controls whether $I_{i,j}^{(k)}$ can be used for the posterior inference of the latent variable $J_i^{(k)}$. Hence, we call our model *Gated Latent Beta Allocation* (GLBA). A graphical illustration of the model is shown in Fig. 6.4.

We now present the mathematical formulation of the model. For $k = 1, \ldots, n$, we generate a set of random variables independently via

$$T_j^{(k)} \quad i.i.d. \quad \sim \quad \text{Bernoulli}(\tau_j), \quad j \in \Omega_k \ , \tag{6.2}$$

$$J_i^{(k)} \quad i.i.d. \quad \sim \quad \text{Beta}(\alpha_i, \beta_i), \quad i \in \Omega_k \ , \tag{6.3}$$

$$I_{i,j}^{(k)} \left| T_j^{(k)}, J_i^{(k)} \right. \quad \sim \quad \begin{cases} \text{Bernoulli}\left(J_i^{(k)}\right) & \text{if } T_j^{(k)} = 1 \\ \text{Bernoulli}(\gamma) & \text{if } T_j^{(k)} = 0 \end{cases} \tag{6.4}$$

where the last random process holds for any $j \in \Omega_k^{-i} := \Omega_k - \{i\}$ and $i \in \Omega_k$ with $k = 1, \ldots, n$, and γ is the rate of agreement by chance if one of i, j turns out to be unreliable. Here $\{I_{i,j}^{(k)}\}$ are observed data.

If a spammer is in the subject pool, his or her reliability parameter τ_i is zero, though others can still agree with his or her answers by chance at rate γ. On the other hand, if one is very reliable yet often provides controversial answers, his reliability τ_i can be one, while he typically disagrees with others, indicated by his high irregularity $\mathbb{E}[J_i^{(k)}] = \frac{\alpha_i}{\alpha_i + \beta_i} \approx 0$. We are interested in finding both types of subjects. However, most of subjects lie in between these two extremes.

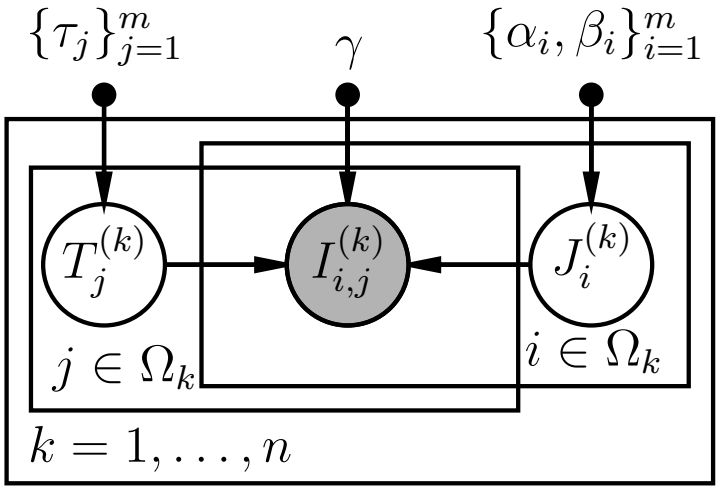


Figure 6.4. Probabilistic graphical model of the proposed Gated Latent Beta Allocation.

As an interesting note, Eq. (6.4) is asymmetric, meaning that $I_{i,j}^{(k)} \neq I_{j,i}^{(k)}$ is possible, a scenario that should never occur by definitions of the two quantities. We propose to achieve symmetry in the final model by using the conditional distribution of $I_{i,j}^{(k)}$ and $I_{j,i}^{(k)}$ given that $I_{i,j}^{(k)} = I_{j,i}^{(k)}$, and call this model the symmetrized model. With details omitted, we state that conditioned on $T_i^{(k)}$, $T_j^{(k)}$, $J_i^{(k)}$, and $J_j^{(k)}$, the symmetrized model is still a Bernoulli distribution:

$$I_{i,j}^{(k)} \sim \text{Bernoulli}\left(H\left(\left(J_i^{(k)}\right)^{T_i^{(k)}} \gamma^{1-T_i^{(k)}}, \left(J_j^{(k)}\right)^{T_j^{(k)}} \gamma^{1-T_j^{(k)}}\right)\right), \tag{6.5}$$

where

$$H(p,q) = \frac{pq}{pq + (1-p)(1-q)} .$$

We tackle the inference and estimation of the asymmetric model for simplicity.

6.2.3 Variational EM

Variational inference is an optimization based strategy for approximating posterior distribution in complex distributions [102]. Since the full posterior is highly intractable, we consider to use variational EM to estimate the parameters $\Theta = (\{\tau_i, \alpha_i, \beta_i\}_{i=1}^m, \gamma)$ [103]. The parameter γ is assumed to be pre-selected by the user and does not need to be estimated. To regularize the other parameters in

estimation, we use the empirical Bayes approach to choose priors. Assume the following priors

$$\tau_i \sim \text{Beta}(\tau_0, 1-\tau_0) \,, \tag{6.6}$$

$$\alpha_i + \beta_i \sim \text{Gamma}(2, s_0) \,. \tag{6.7}$$

By empirical Bayes, τ_0, s_0 are adjusted. For the ease of notations, we define two auxiliary functions $\omega_i^{(k)}(\cdot)$ and $\psi_i^{(k)}(\cdot)$:

$$\omega_i^{(k)}(x) := \sum_{j \in \Omega_k^{\neg i}} x_j I_{i,j}^{(k)}, \quad \psi_i^{(k)}(x) := \sum_{j \in \Omega_k} x_j \,. \tag{6.8}$$

Similarly, we define their siblings

$$\bar{\omega}_i^{(k)}(x) = \omega_i^{(k)}(1-x), \quad \bar{\psi}_i^{(k)}(x) = \psi_i^{(k)}(1-x) \,. \tag{6.9}$$

We also define the auxiliary function $r_j(\cdot)$ as

$$r_j^{(k)}(x) = \prod_{i \in \Omega_k^{\neg j}} \left(\frac{x_i}{\gamma}\right)^{I_{i,j}^{(k)}} \left(\frac{1-x_i}{1-\gamma}\right)^{1-I_{i,j}^{(k)}} . \tag{6.10}$$

Now we define the full likelihood function:

$$L_k(\Theta; T^{(k)}, J^{(k)}, I^{(k)}) := \prod_{j \in \Omega_k} \left((\tau_j)^{T_j^{(k)}} (1-\tau_j)^{1-T_j^{(k)}} \right) \cdot \prod_{i \in \Omega_k} \frac{\left(J_i^{(k)}\right)^{\alpha_i^{(k)}} \left(1-J_i^{(k)}\right)^{\beta_i^{(k)}} \phi_i^{(k)}}{B(\alpha_i, \beta_i)} \,, \tag{6.11}$$

where auxiliary variables simplifying the equations are

$$\begin{aligned} \alpha_i^{(k)} &= \alpha_i + \omega_i^{(k)}\left(T^{(k)}\right) \,, \\ \beta_i^{(k)} &= \beta_i + \psi_i^{(k)} - \omega_i^{(k)}\left(T^{(k)}\right) \,, \\ \phi_i^{(k)} &= \gamma^{\bar{\omega}_i^{(k)}\left(T^{(k)}\right)} (1-\gamma)^{\bar{\psi}_i^{(k)}\left(T^{(k)}\right) - \bar{\omega}_i^{(k)}\left(T^{(k)}\right)} \,, \end{aligned}$$

and $B(\cdot,\cdot)$ is the Beta function. Consequently, assume the prior likelihood is $L_\Theta(\Theta)$, the MAP estimate of Θ is to minimize

$$L(\Theta; T, J, I) := L_\Theta(\Theta) \prod_{k=1}^{n} L_k(\Theta; T^{(k)}, J^{(k)}, I^{(k)}) \,. \tag{6.12}$$

We solve the estimation using variational EM method with a fixed (τ_0, s_0) and varying γ. The idea of variational methods is to approximate the posterior by a factorizable template, whose probability distribution minimizes its KL divergence to the true posterior. Once the approximate posterior is solved, it is then used in the E-step in the EM algorithm as the alternative to the true posterior. The usual M-step is unchanged. Each time Θ is estimated, we adjust prior (τ_0, s_0) to match the mean of the MAP estimates of $\{\tau_i\}$ and $\left\{\frac{\alpha_i + \beta_i}{2}\right\}$ respective until they are sufficiently close.

E-step. We use the factorized Q-approximation with variational principle:

$$p_\Theta\left(T^{(k)}, J^{(k)} \middle| I^{(k)}\right) \approx \prod_{j\in\Omega_k} q^*_{T_j,\Theta}\left(T_j^{(k)}\right) \prod_{i\in\Omega_k} q^*_{J_i,\Theta}\left(J_i^{(k)}\right). \tag{6.13}$$

- Let

$$q^*_{T_j,\Theta}\left(T_j^{(k)}\right) \propto \exp\left(\mathbb{E}_{J,T^{\neg j}}\left[\log L_k\left(\Theta; T^{(k)}, J^{(k)}, I^{(k)}\right)\right]\right) \,, \tag{6.14}$$

  whose distribution can be written as

$$\text{Bernoulli}\left(\frac{\tau_j R_j^{(k)}}{\tau_j R_j^{(k)} + 1 - \tau_j}\right),$$

  where $\log R_j^{(k)} = \mathbb{E}_J\left[\sum_{i\in\Omega_k^{\neg j}} \log\left(r_i^{(k)}(J^{(k)})\right)\right]$. As suggested by Johnson and Kotz [104], the geometric mean can be numerically approximated by

$$R_j^{(k)} \approx \prod_{i\in\Omega_k^{\neg j}} \frac{1}{\alpha_i^{(k)} + \beta_i^{(k)}} \left(\frac{\alpha_i^{(k)}}{\gamma}\right)^{I_{i,j}^{(k)}} \left(\frac{\beta_i^{(k)}}{1-\gamma}\right)^{1-I_{i,j}^{(k)}}, \tag{6.15}$$

  if both $\alpha_i^{(k)}$ and $\beta_i^{(k)}$ are sufficiently larger than 1.

- Let

$$q^*_{J_i,\Theta}(J_i^{(k)}) \propto \exp\left(\mathbb{E}_{T,J^{\neg i}}\left[\log L_k\left(\Theta; T^{(k)}, J^{(k)}, I^{(k)}\right)\right]\right) , \tag{6.16}$$

whose distribution is

$$\text{Beta}(\alpha_i + \omega_i^{(k)}(\tau), \beta_i + \psi_i^{(k)}(\tau) - \omega_i^{(k)}(\tau)) .$$

Given parameter $\tilde{\Omega} = \{\tilde{\tau}_i, \tilde{\alpha}_i, \tilde{\beta}_i\}_{i=1}$, we can compute the approximate posterior expectation of the log likelihood, which reads

$$\begin{aligned}
&\mathbb{E}_{T,J|\tilde{\Theta},I} \log L_k(\Theta; T^{(k)}, J^{(k)}, I^{(k)}) \approx \\
&\text{const} . + \log L_\Theta(\Theta) + \sum_{j\in\Omega_k}\left(\tilde{\tau}_i^{(k)} \log \tau_j + (1-\tilde{\tau}_i^{(k)}) \log(1-\tau_j)\right) + \\
&\sum_{i\in\Omega_k}\left\langle \begin{pmatrix} \alpha_i \\ \beta_i \end{pmatrix}, \frac{\nabla B(\tilde{\alpha}_i^{(k)}, \tilde{\beta}_i^{(k)})}{B(\tilde{\alpha}_i^{(k)}, \tilde{\beta}_i^{(k)})} \right\rangle - \sum_{i\in\Omega_k} \log B(\alpha_i, \beta_i) + \log\gamma \sum_{i\in\Omega_k} \bar{\omega}_i^{(k)}\left(\tilde{\tau}_i^{(k)}\right) + \\
&\log(1-\gamma) \sum_{i\in\Omega_k}\left(\bar{\psi}_i^{(k)}\left(\tilde{\tau}_i^{(k)}\right) - \bar{\omega}_i^{(k)}\left(\tilde{\tau}_i^{(k)}\right)\right),
\end{aligned} \tag{6.17}$$

where relevant statistics are defined as

$$\begin{aligned}
\tilde{\alpha}_i^{(k)} &= \tilde{\alpha}_i + \omega_i^{(k)}(\tilde{\tau}) , \\
\tilde{\beta}_i^{(k)} &= \tilde{\beta}_i + \psi_i^{(k)}(\tilde{\tau}) - \omega_i^{(k)}(\tilde{\tau}) , \text{ and} \\
\tilde{\tau}_i^{(k)} &= \frac{\tilde{R}_i^{(k)}\tilde{\tau}_i}{\tilde{R}_i^{(k)}\tilde{\tau}_i + 1 - \tilde{\tau}_i} .
\end{aligned} \tag{6.18}$$

Remark $B(\cdot,\cdot)$ is the Beta function, and $\tilde{R}_i^{(k)}$ is calculated from approximation Eq. (6.15)

M-step. Compute the partial derivatives of L with respect to α_i and β_i: let Δ_i be the set of images that are labeled by subject i. We set $\partial L/\partial\alpha_i = 0$ and $\partial L/\partial\beta_i = 0$ for each i, which reads

$$\begin{aligned}
&\left(\frac{\alpha_i+\beta_i}{s_0} - \log(\alpha_i+\beta_i)\right)\cdot\begin{pmatrix}1\\1\end{pmatrix} = \sum_{k\in\Delta_i}\frac{\nabla B(\tilde{\alpha}_i^{(k)}, \tilde{\beta}_i^{(k)})}{B(\tilde{\alpha}_i^{(k)}, \tilde{\beta}_i^{(k)})} - \frac{\nabla B(\alpha_i,\beta_i)}{B(\alpha_i,\beta_i)} \\
&= \sum_{k\in\Delta_i}\begin{pmatrix}\Psi(\tilde{\alpha}_i^{(k)}) - \Psi(\tilde{\alpha}_i^{(k)} + \tilde{\beta}_i^{(k)}) \\ \Psi(\tilde{\beta}_i^{(k)}) - \Psi(\tilde{\alpha}_i^{(k)} + \tilde{\beta}_i^{(k)})\end{pmatrix} - |\Delta_i|\cdot\begin{pmatrix}\Psi(\alpha_i) - \Psi(\alpha_i+\beta_i) \\ \Psi(\beta_i) - \Psi(\alpha_i+\beta_i)\end{pmatrix}
\end{aligned} \tag{6.19}$$

where $\Psi(x) \in [\log(x-1), \log x]$ is the Digamma function. The above two equations can be practically solved by Newton-Raphson method with a projected modification (ensuring α, β always are greater than zero).

Compute the derivatives of L with respect to τ_i and set $\partial L/\partial \tau_i = 0$, which reads

$$\tau_i \quad = \quad \frac{1}{|\Delta_i| + 1} \left(\tau_0 + \sum_{k \in \Delta_i} \tilde{\tau}_i^{(k)} \right) . \tag{6.20}$$

Compute the derivatives of L w.r.t. γ and set to zero, which reads

$$\gamma = \frac{\sum_{i \in \Omega_k} \bar{\omega}_i^{(k)}(\tilde{\tau}_i^{(k)})}{\sum_{i \in \Omega_k} \bar{\psi}_i^{(k)}(\tilde{\tau}_i^{(k)})} . \tag{6.21}$$

In practice, the update formula for γ needs not to be used if γ is pre-fixed. See Algorithm 6.1 for details.

6.2.4 The Algorithm

We present our final algorithm to estimate all parameters by knowing the multigraph data $\{I_{i,j}^{(k)}\}$. Our algorithm is designed based on Eqs. (6.19), (6.20), and (6.21). In each EM iteration, there are two loops: one for collecting relevant statistics for each subgraph, and the other for re-computing the parameter estimates for each subject. Please refer to Algorithm 6.1 for details.

Algorithm 6.1 Variational EM algorithm of GLBA

Input: A multi-graph $\{I_{i,j}^k \in \{0,1\}\}_{i,j\in\Omega_k}$, $0 < \gamma < 0.5$

Output: subject parameters $\Theta = (\{(\tau_i, \alpha_i, \beta_i)\}_{i=1}^m, \gamma)$

Initialisation : $\tau_0 = 0.5, \alpha_i = \beta_i = \tau_i = 1.0, i = 1, \ldots, m$
repeat
  for $k = 1$ to n **do**
    compute statistics $\tilde{\alpha}_i^{(k)}, \tilde{\beta}_i^{(k)}, \tilde{\tau}_i^{(k)}$ by Eq. (6.18);
  end for
  for $i = 1$ to m **do**
    solve (α_i, β_i) from Eq. (6.19) (Newton-Raphson);
    compute τ_i by Eq. (6.20);
  end for
  (optional) update γ from Eq. (6.21);
until $\{(\tau_i, \alpha_i, \beta_i)\}_{i=1}^m$ are all converged. **return** Θ

6.3 Experiments

6.3.1 Data Sets

We studied a crowdsourced affective data set acquired from the Amazon Mechanical Turk (AMT) platform [85]. The affective data set is a collection of image stimuli and their affective labels including valence, arousal, dominance and likeness (degree of appreciation). Labels for each image are ordinal: {1, ... , 9} for the first three dimensions, and {1, ..., 7} for the likeness dimension. The study setup and collected data statistics have been detailed in [85], which we describe briefly here for the sake of completeness.

At the beginning of a session, the AMT study host provides the subject brief training on the concepts of affective dimensions. Here are descriptions used for valence, arousal, dominance, and likeness.

- Valence: degree of feeling happy vs. unhappy
- Arousal: degree of feeling excited vs. calm
- Dominance: degree of feeling submissive vs. dominant

- Likeness: how much you like or dislike the image

The questions presented to the subject for each image are given below in exact wording.

- Slide the solid bubble along each of the bars associated with the 3 scales (Valence, Arousal, and Dominance) in order to indicate how you ACTUALLY FELT WHILE YOU OBSERVED THE IMAGE.

- How did you like this image? (Like extremely, Like very much, Like slightly, Neither like nor dislike, Dislike slightly, Dislike very much, Dislike extremely)

Each AMT subject is asked to finish a set of labeling tasks, and each task is to provide affective labels on a single image from a prepared set, called the EmoSet. This set contains around 40,000 images crawled from the Internet using affective keywords. Each task is divided into two stages. First, the subject views the image; and second, he/she provides ratings in the emotion dimensions through a Web interface. Subjects usually spend three to ten seconds to view each image, and five to twenty seconds to label it. The system records the time durations respectively for the two stages of each task and calculates the average cost (at a rate of about 1.4 US Dollars per hour). Around 4,000 subjects were recruited in total. For the experiments below, we retained image stimuli that have received affective labels from at least four subjects. Under this screening, the AMT data have 47,688 responses from 2,039 subjects on 11,038 images. Here, one response refers to the labeling of one image by one subject conducted in one task.

Because humans can naturally feel differently from each other in their affective experiences, there was no gold standard criterion to identify spammers. Such a human emotion data set is difficult to analyze and the quality of data is hard to assess. Among several emotion dimensions, we found that participants were more consistent in the valence dimension. As a reminder, valence is the rated degree of positivity of emotion evoked by looking at an image. We call the variance of the ratings from different subjects on the same image the within-task variance, while the variance of the ratings from all the subjects on all the images the cross-task variance. For valence and likeness, the within-task variance accounts for about 70% of the cross-task variance, much smaller than for the other two dimensions. Therefore, the remaining experiments were focused on evaluating the regularity of image valences in the data.

6.3.2 Baselines for Comparison

We discuss below several baseline methods or models with which we compare our method.

Dawid and Skene [86]. Our method falls into the general category of consensus methods in the literature of statistics and machine learning, where the spammer filtering decision is made completely based on the labels provided by observers. Those consensus methods have been developed along the line of Dawid and Skene [86], and they mainly deal with categorical labels by modeling each observer using a designated confusion matrix. More recent developments of the observer models have been discussed in [94], where a benchmark has shown that the Dawid-Skene method is still quite competitive in unsupervised settings according to a number of real-world data sets for which ground-truth labels are believed to exist albeit unknown. However, this method is not directly applicable to our scenario. To enable comparison with this baseline method, we first convert each affective dimension into a categorical label by thresholding. We create three categories: high, neural, and low, each covering a continuous range of values on the scale. For example, high valence category implies a score greater than a neural score (*i.e.*, 5) by more than a threshold (*e.g.*, 0.5). Such a thresholding approach has been adopted in developing affective categorization systems, *e.g.* [83, 84].

Time duration. In the practice of data collection, the host filtered spammers by a simple criterion—to declare a subject spammer if he spends substantially less time on every task. The labels provided by the identified spammers were then excluded from the data set for subsequent use, and the host also declined to pay for the task. However, some subjects who were declined to be paid wrote emails to the host arguing for their cases. Under this spirit, in our experiments, we form a baseline method that uses the average time duration of each subject to red-flag a spammer.

Filtering based on gold standard examples. A widely used spammer detection approach in crowdsourcing is to create a small set with known ground truth labels and use it to spot anyone who gives incorrect labels. However, such a policy was not implemented in our data collection process because as we argued earlier, there is simply no ground truth for the emotion responses to an image in a general sense. On the other hand, just for the sake of comparison, it seems reasonable to

find a subset of images that evoke such extreme emotions that ground truth labels can be accepted. This subset will then serve the role of gold standard examples. We used our method to retrieve a subset of images which evoke extreme emotions with high confidence (see Section 6.3.7 for confidence score and emotion score calculation). For the valence dimension, we were able to identify at most 101 images with valence score ≥ 8 (on the scale of $1 \ldots 9$) with over 90% confidence and 37 images with valence score ≤ 2 with over 90% confidence. We also looked at those images one by one (as provided in the supplementary materials) and believe that within a reasonable tolerance of doubt those images should evoke clear emotions in the valence dimension. Unfortunately, only a small fraction of subjects in our pool have labeled at least one image from this "gold standard" subset. Among this small group, their disparity from the gold standard enables us to find three susceptible spammers. To see whether these three susceptible spammers can also be detected by our method, we find that their reliability scores $\tau \in [0, 1]$ are $0.11, 0.22, 0.35$ respectively. In Fig. 6.9, we plot the distribution of τ of the entire subject pool. These three scores are clearly on the low end with respect to the scores of the other subjects. Thus the three spammers are also assessed to be highly susceptible by our model.

In summary, while we were able to compare our method with the first two baselines quantitatively, with results to be presented shortly, comparison with the third baseline is limited due to the way the AMT data were collected [85].

6.3.3 Model Setup

Since our hypotheses included a random agreement ratio γ that is pre-selected, we adjusted the parameter γ from 0.3 to 0.48 to see empirically how it affects the result in practice.

Fig. 6.5 depicts how the reliability parameter τ varies with γ for different workers in our data set. Results are shown for the top 15 users who provided the most numbers of ratings. Generally speaking, a higher γ corresponds to a higher chance of agreement between workers purely out of random. From the figure, we can see that a worker providing more ratings is not necessarily more reliable. It is quite possible that some workers took advantage of the AMT study to earn monetary compensation without paying enough attention to the actual questions.

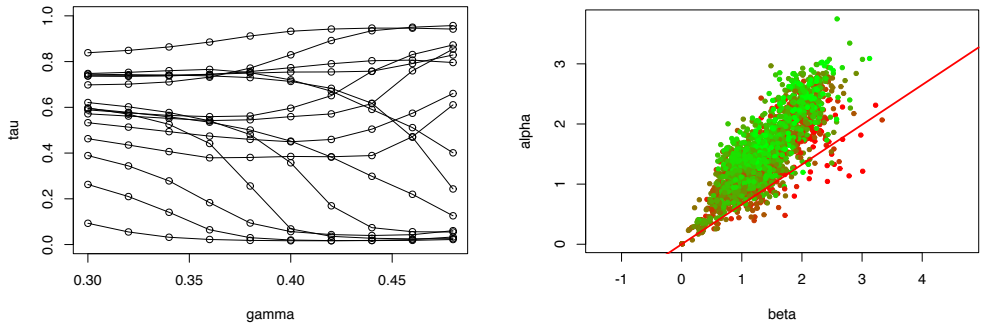


Figure 6.5. Left: Reliability scores versus $\gamma \in [0.3, 0.48]$ for the top 15 users who provided the most numbers of ratings. Right: Visualization of the estimated regularity parameters of each worker at a given γ. Green dots are for workers with high reliability and red dots for low reliability. The slope of the red line equals γ.

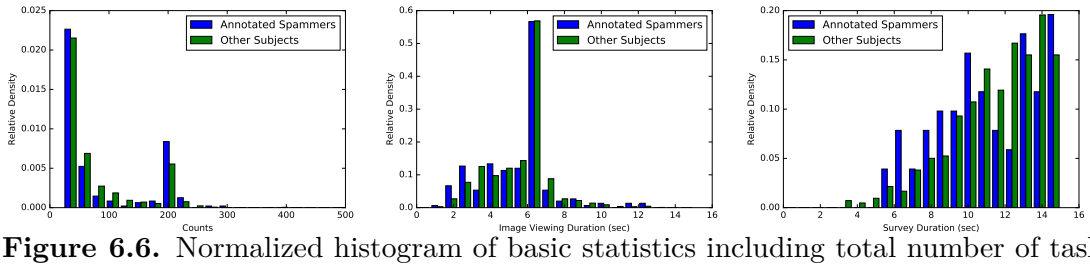


Figure 6.6. Normalized histogram of basic statistics including total number of tasks completed and average time duration spent at each of the two stages per task.

In Table 6.2, we demonstrate the valence, arousal, and dominance labels for two categories of subjects. On the top, the first category contains susceptible spammers with low estimated reliability parameter τ; and on the bottom, the second category contains highly reliable subjects with high values of τ. Each subject takes one row. For the convenience of visualization, we represent the three-dimensional emotion scores given to any image by a particular color whose RGB values are mapped from the values in the three dimensions respectively. The emotion labels for every image by one subject are then condensed into one color bar. The labels provided by each subject for all his images are then shown as a palette in one row. For clarity, the color bars are sorted in lexicographic order of their RGB values. One can clearly see that those labels given by the subjects from these two categories exhibit quite different patterns. The palettes of the susceptible spammers are more extreme in terms of saturation or brightness. The abnormality of label distributions of the first category naturally originates from the fact that spammers intended to label the

τ_i	α_i	β_i	reported emotions (sorted)
0.19	1.17	2.43	
0.08	0.75	2.20	
0.08	1.16	2.50	
0.09	0.67	1.70	
0.03	0.94	1.90	
0.17	0.72	1.47	
0.06	1.14	2.50	
0.17	0.86	1.79	
0.04	1.01	2.63	
0.03	1.08	2.84	
0.92	2.29	1.49	
0.94	2.55	1.98	
0.95	2.61	1.68	
0.92	2.40	1.66	
0.91	2.21	1.40	
0.92	2.45	1.97	
0.93	2.38	1.69	
0.93	1.76	1.40	
0.91	2.44	1.86	
0.92	2.30	1.85	
0.92	2.45	1.82	
0.91	1.64	1.29	
0.90	1.68	1.12	
0.91	2.72	2.22	

Table 6.2. Oracles in the AMT data set. Upper: malicious oracles whose α_i/β_i is among the lowest 30, meanwhile $|\Delta_i|$ is greater than 10. Lower: reliable oracles whose τ_i is among the top 30, meanwhile $\alpha_i/\beta_i > 1.2$. Their reported emotions are visualized by RGB colors. The estimates of Θ is based on the valence dimension.

data by exerting the minimal efforts and without paying attention to the questions.

6.3.4 Basic Statistics of Manually Annotated Spammers

For each subject in the pool, by observing all his or her labels in different emotion dimensions, there was a reasonable chance of spotting abnormality solely by visualizing the distribution. If one were a spammer, it often happened that his or her labels were highly correlated, skewed or deviated in an extreme manner from a neural emotion along different dimensions. In such cases, it was possible to manually exclude his or her responses from the data due to his or her high susceptibility. We applied this same practice to identifying highly susceptible subjects from the pool. We found about 200 susceptible participants.

We studied several basic statistics of this subset in comparison with the whole population: total number of tasks completed, average time duration spent on image viewing and survey per task. The histograms of these quantities are plotted in Fig. 6.6. One can see that the annotated spammers did not necessarily spend less time or finish fewer tasks than the others, and the time duration has shown only marginal sensitivity to those annotated spammers (See Fig. 6.6). The figures demonstrate that those statistics are not effective criteria for spammer filtering.

We will use this subset of susceptible subjects as a "pseudo-gold standard" set for quantitative comparisons of our method and the baselines in the subsequent studies. As explained previously in 6.3.2, other choices of constructing a gold standard set either conflict the high variation nature of emotion responses or yield only a tiny (of size three) set of spammers.

6.3.5 Top-K Precision Performance in Retrieving the Real Spammers

We conducted experiments on each affective dimension, and evaluated whether the subjects with the lowest estimated τ were supposed to be real spammers according to the "pseudo-gold standard" subset constructed in Section 6.3.4. Since there was no gold standard to correctly classify whether one subject was truly a spammer or not, we have been agnostic here. Based on that subset, we were able to partially evaluate the top-K precision in retrieving the real spammers, especially the most susceptible ones.

Specifically, we computed the reliability parameter τ for each subject and chose the K subjects with the lowest values as the most susceptible spammers. Because

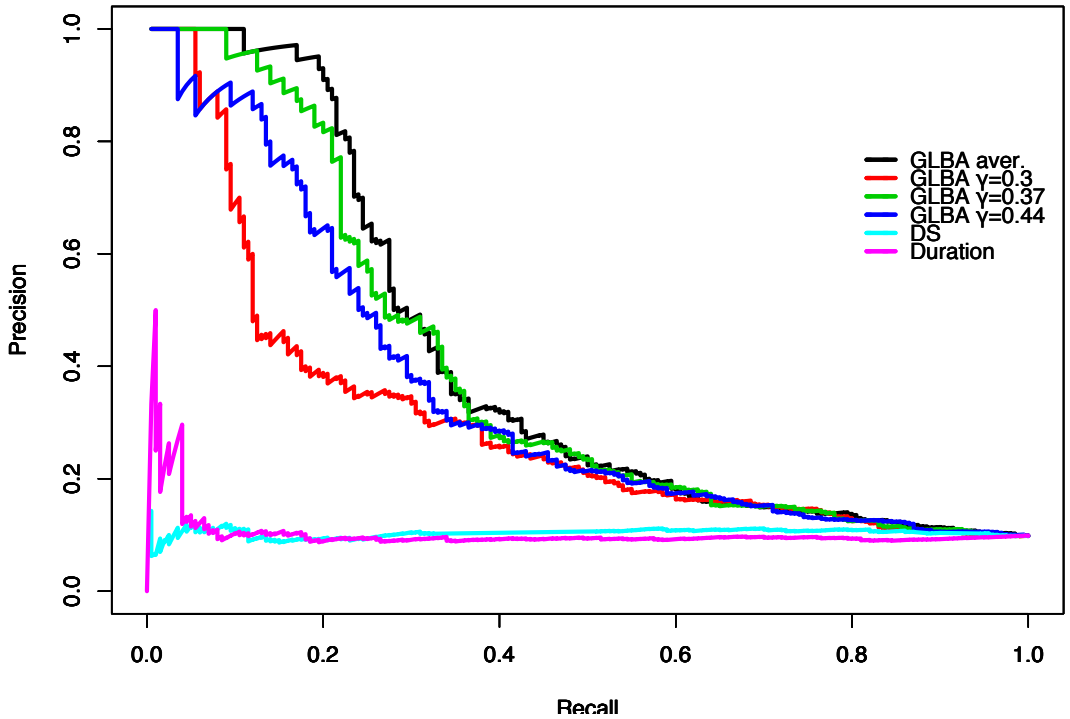


Figure 6.7. The agnostic Precision-Recall curve (by valence) based on manually annotated spammers. The top 20, top 40 and top 60 precision is 100%, 95%, 78% respectively (black line). It is expected that precision drops quickly with increasing recalls, because the manually annotation process can only identify a special type of spammers, while other types of spammers can be identified by the algorithm. The PR curves at $\gamma = 0.3, 0.37, 0.44$ are also plotted. Two baselines are compared: the Dawid and Skene (DS) approach and the time duration based approach.

τ depends on the random agreement rate γ, we computed τ's using 10 values of γ evenly spaced out over interval $[0.3, 0.48]$. The average value of τ was then used for ranking. The Precision Recall Curves are shown in Fig. 6.7. Our method achieves high top-K precision by retrieving the most susceptible subjects from the pool according to the average τ. In particular, the top-20 precision is 100%, the top-40 precision is 95%, and the top-60 precision is 78%. Clearly, our algorithm has yielded results well aligned with the human judgment on the most susceptible ones. In Fig. 6.7, we also plot Precision Recall Curves by fixing γ to $0.3, 0.37, 0.44$ and using the corresponding τ. The result at $\gamma = 0.37$ is better than the other two across recalls, indicating that a proper level of the random agreement rate can be important for achieving the best performance. The two baseline methods are clearly not competitive in this evaluation. The Dawin-Skene method [86], widely

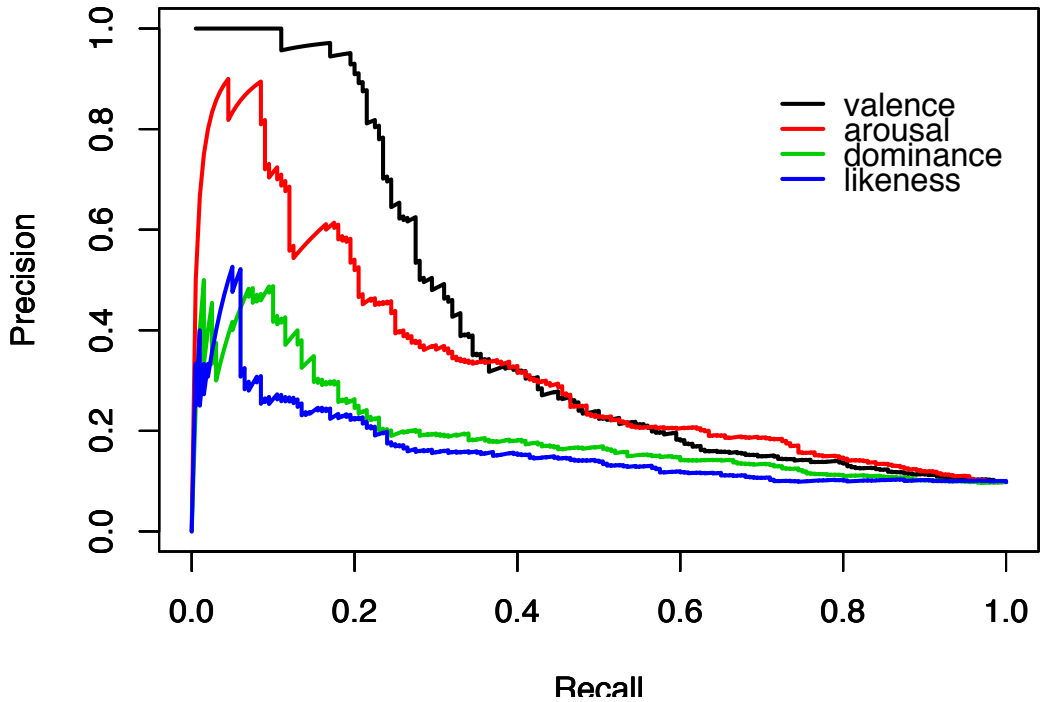


Figure 6.8. The agnostic Precision-Recall curve based on manually annotated spammers computed from different affective dimensions: valence, arousal, dominance, and likeness.

used in processing crowdsourced data with objective ground truth labels, drops quickly to a remarkably low precision even at a low recall. The time duration method, used in the practice of AMT host, is better than the Dawin-Skene method, yet substantially worse than the performance of our method.

We also tested this same method of identifying spammers using affective dimensions other than valence. As shown in Fig. 6.8, the two most discerning dimensions were valence and arousal. It is not surprising that people can reach relatively higher consensus when rating images by these two dimensions than by dominance or likeness. Dominance is much more likely to draw on evidence from context and social situation in most circumstances and hence less likely to have its nature determined to a larger extent by the stimulus itself.

6.3.6 Recall Performance in Retrieving the Simulated Spammers

The evaluation of top-K precision was limited in two respects: (1) the susceptible subjects were identified because we could clearly observe their abnormality in terms

of the multivariate distribution of provided labels. If the participant labeled the data by acting exactly the same as the distribution of the population, we could not manually identify him/her using the aforementioned methodology. (2) We still need to determine if one is a spammer, how likely we are to spot him/her.

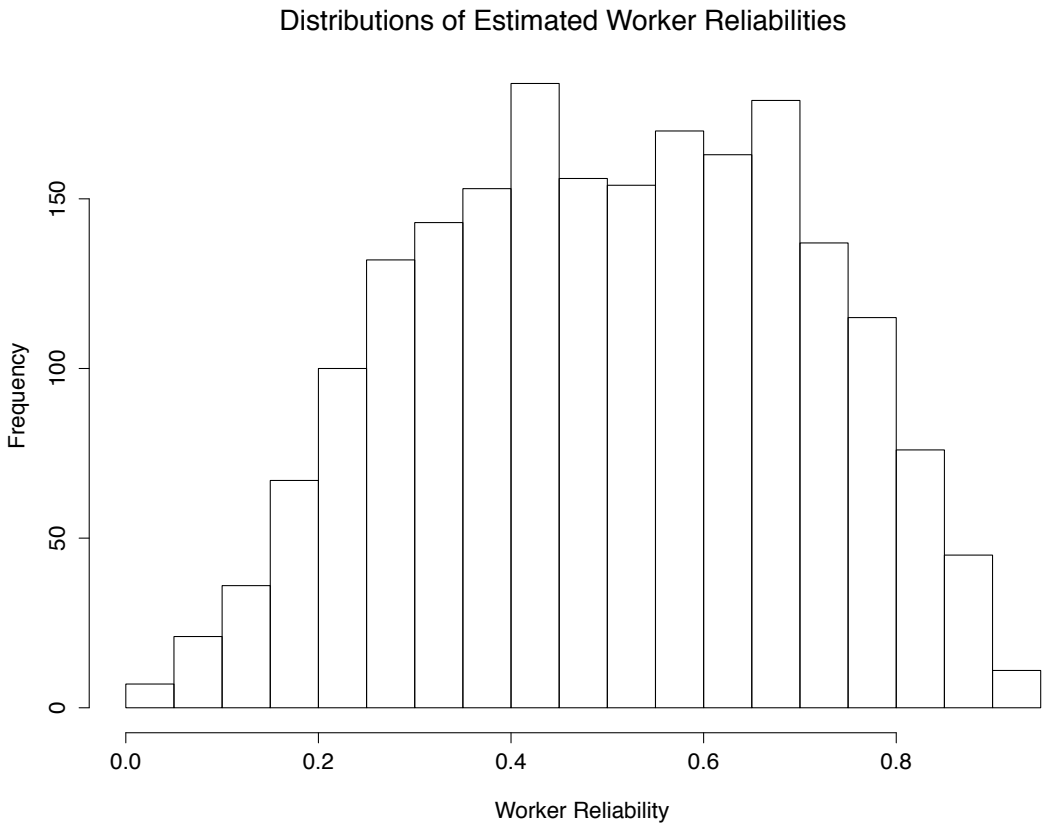
In this study, we simulated several highly "intelligent" spammers, who labeled the data by exactly following the label distribution of the whole population. Every time, we generated 10 spammers, who randomly labeled 50 images. The labels of simulated spammers were not overlapping. We mixed those labels of the simulated spammers with the existing data set, and then conducted our method again to determine how accurate our approach was with respect to finding the simulated spammers. We repeated this process 10 times in order to estimate the τ distribution of the simulated spammers. Results are reported Fig. 6.9. We drew the histogram of the estimated reliability of all real workers and compared them to the estimated reliability of simulated spammers (in the table included in Fig. 6.9). We noted that more than half of the simulated spammers were identified as highly susceptible based on the τ estimation (≤ 0.2), and none of them were supposed to have a high reliability score (≥ 0.6). This result validates that our method is robust enough to spot the "intelligent" spammers, even if they disguise themselves as random labelers within a population.

6.3.7 Qualitative Comparison Based on Controversial Examples

To re-rank the emotion dimensions and likenesses of stimuli with the reliability of the subject accounted for, we adopted the following formula to find the stimuli with "reliably" highest ratings. Assume each rating $a_i \in [0, 1]$. We define the following to replace the usual average:

$$b_k := \underbrace{\frac{\sum_{i \in \Omega_k} \tau_i a_i^{(k)}}{\sum_{i \in \Omega_k} \tau_i}}_{\text{est. score}} \cdot \underbrace{\left(1 - \prod_{i \in \Omega_k} (1 - \tau_i)\right)}_{\text{confidence}}, \tag{6.22}$$

where $\left(1 - \prod_{i \in \Omega_k}(1 - \tau_i)\right) \in [0, 1]$ is the *cumulative confidence score* for image k. This adjusted rating b_k not only allows more reliable subjects to play a bigger role via the weighted average (the first term of the product) but also modulates the weighted average by the cumulative confidence score for the image. Similarly, in



Ranges	≤ 0.2	0.2∼0.4	0.4∼0.6	0.6∼0.8	$\geq$0.8
Counts	54	34	12	0	0

Figure 6.9. The histogram distribution of estimated worker reliabilities τ and statistics of simulated spammers based on 10 repeated runs, each with 10 spammers injected.

order to find those with "reliably" lowest ratings, we replace $a_i^{(k)}$ with $(1 - a_i^{(k)})$ in the above formula and then still seek for the images with the highest b_k's.

If b_k is higher than a neutral level, then the emotional response to the image is considered high. Fig. 6.10 shows the histogram of image confidence scores estimated by our method. More than 85% of images had acquired a sufficient number of quality labels. To obtain a qualitative sense of the usefulness of the reliability parameter τ, we compared our approach with the simple average-and-rank scheme by examining controversial image examples according to each emotion dimension. Here, being controversial means the assessment of the average emotion response for an image differs significantly between the methods. Despite the variability of human nature, the majority of the population were quite likely to reach consensus

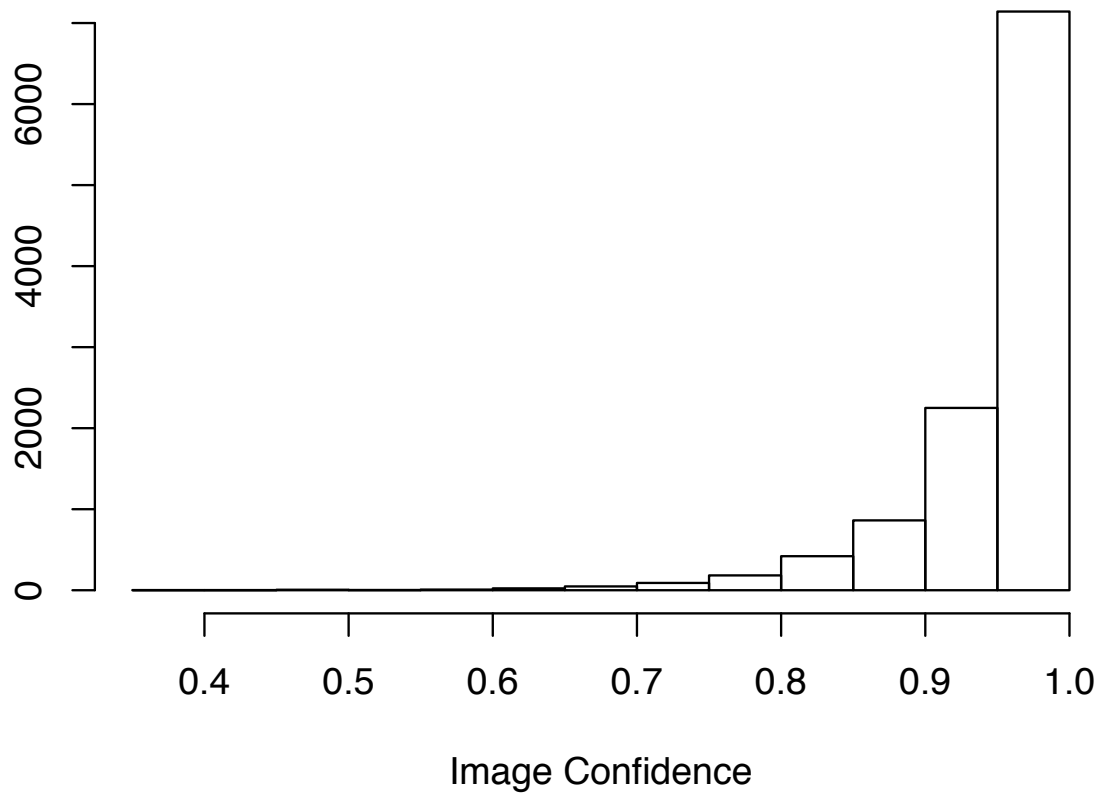


Figure 6.10. The histogram of image confidences estimated based on our method. About 85% of images have a confidence scores higher than 90%.

for a portion of the stimuli. Therefore, this investigation is meaningful. In Fig. 6.2 and Fig. 6.3, we show example image stimuli that were recognized to clearly deviate from neutral emotions by one method but not agreed upon by the other. We skipped stimuli images that were fear inducing, visually annoying or improper. Interested readers can see the complete results in the supplementary material.

6.3.8 Cost/Overhead Analysis

There is an inevitable trade-off between the quality of the labels and the average cost of acquiring them when screening is applied based on reliability. If we set a higher standard for reliability, the quality of the labels retained tends to improve but we are left with fewer labels to use. It is interesting to visualize the trade-off quantitatively. Let us define overhead numerically as the number of labels removed from the data set when quality control is imposed; and let the threshold on either

subject reliability or image confidence used to filter labels be the index for label quality. We obtained what we call *overhead curve* in Fig. 6.11. On the left plot, the result is based on filtering subjects with reliability scores below a threshold (all labels given by such subjects are excluded); on the right, it is based on filtering images with confidence scores below a threshold. As shown by the plots, if either the labels from subjects with reliability scores below 0.3 are discarded or those for images with confidence scores below 90% are discarded, roughly 10,000 out of 47,688 labels are deemed unusable. At an even higher standard, e.g., subject reliability $\geq .5$ or image confidence level $\geq 95\%$, around half of the labels will be excluded from the data set. Although this means the average per label cost is doubled at the stringent quality standard, we believe the screening is worthwhile in comparison with analysis misled by wrong data. In a large-scale crowdsource environment, it is simply impractical to expect all the subjects to be fully serious. This contrasts starkly with a well-controlled lab environment for data collection. In a sense, post-collection analysis of data to ensure quality is unavoidable. It is indeed a matter of which analysis should be applied.

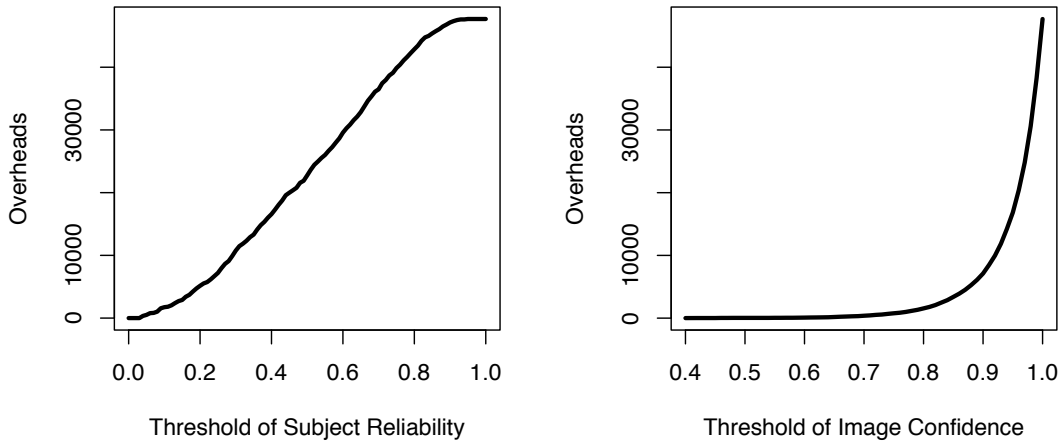


Figure 6.11. Left: Overhead curve based on subject filtering; Right: overhead curve based on image filtering. The overhead is quantified by the number of labels discarded after filtering.

6.4 Discussions

Underlying Principles: Our approach to assess the reliability of crowdsourced affective data deviates fundamentally from the standard approaches much concerned with hunting for "ground truth" emotion stimulated by an image. An individual's emotion response is expected to be naturally different because it depends on subjective opinions rooted in the individual's lifetime exposure to images and concepts, a topic having been pursued long in the literature of social psychology. The new principle we adopted here focuses on the relational knowledge about the ratings of the subjects. Our analysis steps away from the use of "ground truth" by recasting the data as relational quantities.

As pointed out by a reviewer, such a relational perspective may be intrinsic in human cognition, going beyond our specific problem here. For instance, the same spirit of exploiting relationships has already appeared in studies to understand linguistic learning. Gentner [105, 106] proposed that one should understand linguistic learning in a relational way. Instead of assuming there are well-formed abstract language concepts to grasp, the human's cognitive ability often starts from analogical processing based on examples of a concept, and then utilizes the symbolic systems (languages) to reinforce and guide the learning, and to facilitate memory of the acquired concepts. The relationships among the examples and the abstract concept play a role in learning hand in hand, refining recursively the understanding of each other. The whole process is an interlocked and repeated improvement of one side assisted by the other. In a similar fashion, our system improves its assessment about which images evoke highly consensus emotion responses and which subjects are reliable. At the beginning, the lack of either kind of information obscures the truth about the other. Or equivalently, knowing either makes the understanding of the other easy. This is a chicken-and-egg situation. Like the proposed way of learning languages, our system pulls out of the dilemma by recursively enhancing the understanding of one side conditioned on what has been known about the other.

Results: We found that the crowdsourced affective data we examined are particularly challenging for the conventional school of observer models, developed along the line of Dawid and Skene [86]. We identified two major reasons. First, each image in our data set has a much smaller number of observers, compared with what

are typically studied in the benchmarks [94]. In our data set, most images were only labeled by 4 to 8 subjects, while many existing benchmark data sets have tens of subjects per task. Second, a more profound reason is that most images do not have a ground truth affective label at the first place. This can render ineffective many statistical methods which model the user-task confusion matrix and hence count on the existence of "true" labels and the fixed characteristics of uncertainty in responses (assumptions A1 and A2).

Our experiments demonstrate that valence and arousal are the two most effective dimensions that can be used to analyze the reliability of subjects. Although subjects may not reach a consensus at local scales (say, an individual task) because the emotions are inherently subjective, consensus at a global scale can still be well justified.

Usage Scenarios: We would like to articulate on the scenarios under which our method or other traditional approaches (*e.g.*, those described in Section 6.3.2) are more suitable.

First, our method is not meant to replace traditional approaches that add control factors at the design stage of the experiments, for example, recording task completion time, and testing subjects with examples annotated with gold standard labels. Those methods are effective at identifying extremely careless subjects. But we argue that the reliability of a subject is often not a matter of yes or no, but can take a continum of intermediate levels. Moreover, consensus models such as Dawid-Skene methods require that each task is assigned to multiple annotators.

Second, our method can be integrated with other approaches so as to collect data most efficiently. Traditional heuristic approaches require the host to come up with a number of design questions or procedures effective for screening spammers before executing the experiments, which can be a big challenge especially for affective data. In contrast, the consensus models support post analyses of collected data and have no special requirement for the experimental designs. This suggests we may use a consensus model to carry out a pilot study which then informs us how to best design the data collection procedure.

Third, as a new method in the family of consensus models, our approach is unique in terms of its fundamental assumptions, and hence should be utilized in quite different scenarios than the other models. Methods based on modeling

confusion matrix are more suitable for aggregating binary and categorical labels, while the agreement-based methods (ours included) are more suitable for continuous and multi-dimensional labels (or more complicated structures) that normally have no ground truth. The former are often evaluated quantitatively by how accurately they estimate the true labels [94], while the latter are evaluated directly by how effectively they identify unreliable annotators, a perspective barely touched in the existing literature.

Limitations and Future Work: Despite the fact that we did not assume A1 or A2 and approached the problem of assessing the quality of crowdsourced data form an unusual angle, there are interesting questions left about the statistical model we employed.

- Some choices of parameters in the model are quite heuristic. The usage of our model requires pre-set values for certain parameters, *e.g.*, γ, but we have not found theoretically pinned-down guidelines on how to choose those parameters. As a result, it is always subjective to some extent to declare a subject spammer. The ranking of reliability of subjects seems easier to accept. Where the cutoff should be will involve some manual checking on the result or will be determined by some other factors such as the desired cost of acquiring a certain amount of data.

- Although we have made great efforts to design various measures to evaluate our method, struggling to get around the issue of lacking an objective gold standard (its very existence has been questioned), these measures have limitations in one way or the other, as discussed in Section 6.3. We feel that due to the subjective nature of emotion responses to images, there is no simple and quick solution to this. The ultimate test of the method has to come from its usage in practice and a relatively long-term evaluation from the real-world.

- The effects of subgroup consistency, though varied from task to task, were random effects. We constructed the model this way to stretch its applicability because the number of responses collected per task in our empirical data was often small. Some related approaches (*e.g.* [93]) propose to estimate a difficulty/consistency parameter for each task, but often require a relatively

large number of annotators per task. Which kind of probabilistic assumptions is more accurate or works better calls for future exploration.

- Only one "major" reliable mode was assumed at one time, and hereafter only the regularities conditioned on this mode are estimated. In another word, all the reliable users are assumed to behave consistently. One may ask whether there exist subgroups of reliable users who behave consistently within a group but differ across groups for reasons such as different demographic backgrounds. In our current model, if such "minor" reliable mode exists in a population, these subjects may be absorbed into the spammer group. Our model implicitly assumes that diversity in demography or in other aspects does not cause influential differences in emotion responses. Because of this, our method in dealing with culturally sensitive data is not well justified.

Experimentally our method is only evaluated on one particular large data set [85]. Evaluations on other affective data sets (when publicly available) are of interest.

We have focused on the post analysis of collected data. As a future direction, it is of interest to examine the capacity of our approach to reduce time and cost in the practice of crowdsourcing using A/B test. We hereby briefly discuss an online heuristic strategy to dynamically allocate tasks to more reliable subjects. Recall that our model has two sets of parameters: parameter τ_i indicating the reliability of subjects and parameter $\alpha_i; \beta_i$ capturing the regularity. We can use the variance of distribution $\text{Beta}(\alpha_i, \beta_i)$ to determine how confident we are with the estimation of τ_i. For subject i, if the variance of $\text{Beta}(\alpha_i, \beta_i)$ is smaller than a threshold while τ_i is below a certain percentile, this subject is considered *confidently* unreliable and he/she may be excluded from the future subject pool.

In this chapter, we developed a probabilistic model, namely Gated Latent Beta Allocation, to analyze the off-line consensus for crowdsourced affective data. Compared to the usual crowdsourcing settings, where reliable workers are supposed to have consensus, the consensus analysis of affective data is more challenging because of the innate variation in emotion responses even out of true feelings. To overcome this difficulty, our model estimates the reliability of subjects by exploiting the agreement relationships between their ratings at a global scale. The experiments show that the relational data based on the valence of human responses are more effective than the other emotion dimensions for identifying spammer subjects. By

evaluating and comparing the new method with some standard methods in multiple ways, we find that the results have demonstrated clear advantages and the system seems ready for use in practice.

Channel Pruning of Convolution Layers

A.1 Introduction

Not all computations in a deep neural network are of equal importance. In a typical deep learning pipeline, an expert crafts a neural architecture, which is trained using a prepared dataset. The success of training a deep model often requires trial and error, and such loop usually has little control on prioritizing the computations happening in the neural network. Recently researchers started to develop model-simplification methods for convolutional neural networks (CNNs), bearing in mind that some computations are indeed non-critical or redundant and hence can be safely removed from a trained model without substantially degrading the model's performance. Such methods not only accelerate computational efficiency but also possibly alleviate the model's overfitting effects.

Discovering which subsets of the computations of a trained CNN are more reasonable to prune, however, is nontrivial. Existing methods can be categorized from either the *learning perspective* or from the *computational perspective*. From the learning perspective, some methods use a data-independent approach where the training data does not assist in determining which part of a trained CNN should be pruned, *e.g.* [107] and [108], while others use a data-dependent approach through typically a joint optimization in generating pruning decisions, *e.g.*, [109] and [110]. From the computational perspective, while most approaches focus on setting the

The work presented in this section has been published in the form of a research paper: Jianbo Ye, Xin Lu, Zhe Lin, James Z. Wang, "Rethinking the Smaller-Norm-Less-Informative Assumption in Channel Pruning of Convolution Layers," Proceedings of International Conference on Learning Representations, April 2017.

dense weights of convolutions or linear maps to be structured sparse, we propose here a method adopting a new conception to achieve in effect the same goal.

Instead of regarding the computations of a CNN as a collection of separate computations sitting at different layers, we view it as a network flow that delivers information from the input to the output through different channels across different layers. We believe saving computations of a CNN is not only about reducing what are calculated in an individual layer, but perhaps more importantly also about understanding how each channel is contributing to the entire information flow in the underlying passing graph as well as removing channels that are less responsible to such process. Inspired by this new conception, we propose to design a "gate" at each channel of a CNN, controlling whether its received information is actually sent out to other channels after processing. If a channel "gate" closes, its output will always be a constant. In fact, each designed "gate" will have a prior intention to close, unless it has a "strong" duty in sending some of its received information from the input to subsequent layers. We find that implementing this idea in pruning CNNs is unsophisticated, as will be detailed in Sec A.4.

Our method neither introduces any extra parameters to the existing CNN, nor changes its computation graph. In fact, it only introduces marginal overheads to existing gradient training of CNNs. It also possess an attractive feature that one can successively build multiple compact models with different inference performances in a single round of resource-intensive training (as in our experiments). This eases the process to choose a balanced model to deploy in production. Probably, the only applicability constraint of our method is that all convolutional layers and fully-connected layer (except the last layer) in the CNN should be batch normalized [111]. Given batch normalization has becomes a widely adopted ingredient in designing state-of-the-art deep learning models, and many successful CNN models are using it, we believe our approach has a wide scope of potential impacts.[1]

In this paper, we start from rethinking a basic assumption widely explored in existing channel pruning work. We point out several issues and gaps in realizing this assumption successfully. Then, we propose our alternative approach, which works around several numerical difficulties. Finally, we experiment our method

[1]For convolution layer which is not originally trained with batch normalization, one can still convert it into a "near equivalent" convolution layer with batch normalization by removing the bias term b and properly setting $\gamma = \sqrt{\sigma + \epsilon}$, $\beta = b + \mu$, where σ and μ are estimated from the outputs of the convolution across all training samples.

across different benchmarks and validate its usefulness and strengths.

A.2 Related Work

Reducing the size of neural network for speeding up its computational performance at inference time has been a long-studied topic in the communities of neural network and deep learning. Pioneer works include Optimal Brain Damage [112] and Optimal Brain Surgeon [113]. More recent developments focused on either reducing the structural complexity of a provided network or training a compact or simplified network from scratch. Our work can be categorized into the former, thus the literature review below revolves around reducing the structural complexity.

To reduce the structural complexity of deep learning models, previous work have largely focused on sparsifying the weights of convolutional kernels or the feature maps across multiple layers in a network [109, 110]. Some recent efforts proposed to impose structured sparsity on those vector components motivated from the implementation perspective on specialized hardware [114–117]. Yet as argued by [118], regularization-based pruning techniques require per layer sensitivity analysis which adds extra computations. Their method relies on global rescaling of criteria for all layers and does not require sensitivity estimation, a beneficial feature that our approach also has. To our knowledge, it is also unclear how widely useful those works are in deep learning. In Section A.3, we discuss in details the potential issues in regularization-based pruning techniques potentially hurting them being widely applicable, especially for those that regularize high-dimensional tensor parameters or use magnitude-based pruning methods. Our approach works around the mentioned issues by constraining the anticipated pruning operations only to batch-normalized convolutional layers. Instead of posing structured sparsity on kernels or feature maps, we enforce sparsity on the scaling parameter γ in batch normalization operator. This blocks the sample-wise information passing through part of the channels in convolution layer, and in effect implies one can safely remove those channels.

A recent work by [119] used a similar technique as ours to remove unimportant residual modules in ResNet by introducing extra scaling factors to the original network. However, some optimization subtleties as to be pointed out in our paper were not well explained. Another recent work called *Network-Slimming* [120] also

aims to sparsify the scaling parameters of batch normalization. But instead of using off-the-shelf gradient learning like theirs, we propose a new algorithmic approach based on ISTA and rescaling trick, improving robustness and speed of the undergoing optimization. In particular, the work of [120] was able to prune VGG-A model on ImageNet. It is unclear how their work would deal with the γ-W rescaling effect and whether their approach can be adopted to large pre-trained models, such as ResNets and Inceptions. We experimented with the pre-trained ResNet-101 and compared to most recent work that were shown to work well with large CNNs. We also experimented with an image segmentation model which has an inception-like module (pre-trained on ImageNet) to locate foreground objects.

A.3 Rethinking the Smaller-Norm-Less-Informative Assumption

In most regularized linear regressions, a large-norm coefficient is often a strong indicator of a highly informative feature. This has been widely perceived in statistics and machine learning communities. Removing features which have a small coefficient does not substantially affect the regression errors. Therefore, it has been an established practice to use tractable norm to regularize the parameters in optimizing a model and pick the important ones by comparing their norms after training. However, this assumption is not unconditional. By using Lasso or ridge regression to select important predictors in linear models, one always has to first normalize each predictor variable. Otherwise, the result might not be explanatory. For example, ridge regression penalizes more the predictors which has low variance, and Lasso regression enforces sparsity of coefficients which are already small in OLS. Such normalization condition for the right use of regularization is often unsatisfied for nonconvex learning. For example, one has to carefully consider two issues outlined below. We provides these two cases to exemplify how regularization could fail or be of limited usage. There definitely exist ways to avoid the described failures.

Model Reparameterization. In the first case, we show that it is not easy to have fine-grained control of the weights' norms across different layers. One has to

either choose a uniform penalty in all layers or struggle with the reparameterization patterns. Consider to find a deep linear (convolutional) network subject to a least square with Lasso: for $\lambda > 0$,

$$\min_{\{W_i\}_{i=1}^{2n}} \mathbb{E}_{(x,y)\sim\mathcal{D}} \|W_{2n} * \ldots * W_2 * W_1 * x - y\|^2 + \lambda \sum_{i=1}^{n} \|W_{2i}\|_1 \ .$$

The above formulation is not a well-defined problem because for any parameter set $\{W_i\}_{i=1}^{2n}$, one can always find another parameter set $\{W_i'\}_{i=1}^{2n}$ such that it achieves a smaller total loss while keeping the corresponding l_0 norm unchanged by actually setting

$$W_i' = \alpha W_i, i = 1, 3, \ldots, 2n-1 \text{ and } W_i' = W_i/\alpha, i = 2, 4, \ldots, 2n \ ,$$

where $\alpha > 1$. In another word, for any $\epsilon > 0$, one can always find a parameter set $\{W_i\}_{i=1}^{2n}$ (which is usually non-sparse) that minimizes the first least square loss while having its second Lasso term less than ϵ.

We note that gradient-based learning is highly inefficient in exploring such model reparameterization patterns. In fact, there are some recent discussions around this [121]. If one adopts a pre-trained model, and augments its original objective with a new norm-based parameter regularization, the new gradient updates may just increase rapidly or it may take a very long time for the variables traveling along the model's reparameterization trajectory. This highlights a theoretical gap questioning existing sparsity-inducing formulation and actual computational algorithms whether they can achieve widely satisfactory parameter sparsification for deep learning models.

Transform Invariance. In the second case, we show that batch normalization is not compatible with weight regularization. The example is penalizing l_1- or l_2-norms of filters in convolution layer which is then followed by a batch normalization: at the l-th layer, we let

$$x^{l+1} = \max\{\gamma \cdot \mathrm{BN}_{\mu,\sigma,\epsilon}(W^l * x^l) + \beta, 0\},$$

where γ and β are vectors whose length is the number of channels. Likewise, one can clearly see that any uniform scaling of W^l which changes its l_1- and l_2-norms would have no effects on the output x^{l+1}. Alternatively speaking, if one is interested

in minimizing the weight norms of multiple layers together, it becomes unclear how to choose proper penalty for each layer. Theoretically, there always exists an optimizer that can change the weight to one with infinitesimal magnitude without hurting any inference performance. As pointed by one of the reviewers, one can tentatively avoid this issue by projecting the weights to the surface of unit ball. Then one has to deal with a non-convex feasible set of parameters, causing extra difficulties in developing optimization for data-dependent pruning methods. It is also worth noting that some existing work used such strategy in a layer-by-layer greedy way [107, 108].

Based on this discussion, many existing works which claim to use Lasso, group Lasso (e.g. [110, 114]), or thresholding (e.g. [118]) to enforce parameter sparsity have some theoretical gaps to bridge. In fact, many heuristic algorithms in neural net pruning actually do not naturally generate a sparse parameterized solution. More often, thresholding is used to directly set certain subset of the parameters in the network to zeros, which can be problematic. The reason is in essence around two questions. First, by setting parameters less than a threshold to zeros, will the functionality of neural net be preserved approximately with certain guarantees? If yes, then under what conditions? Second, how should one set those thresholds for weights across different layers? Not every layer contributes equally in a neural net. It is expected that some layers act critically for the performance but only use a small computation and memory budget, while some other layers help marginally for the performance but consume a lot resources. It is naturally more desirable to prune calculations in the latter kind of layers than the former.

In contrast with these existing approaches, we focus on enforcing sparsity of a tiny set of parameters in CNN — scale parameter γs in all batch normalization. Not only placing sparse constraints on γ is simpler and easier to monitor, but more importantly, we have two strong reasons:

1. Every γ always multiplies a normalized random variable, thus the channel importance becomes comparable across different layers by measuring the magnitude values of γ;

2. The reparameterization effect across different layers is avoided if its subsequent convolution layer is also batch-normalized. In other words, the impacts from

the scale changes of γ parameter are independent across different layers.

Nevertheless, our current work still falls short of a strong theoretical guarantee. We believe by working with normalized feature inputs and their regularized coefficients together, one is closer to a more robust and meaningful approach. Sparsity is not the goal, but to find less important channels using sparsity inducing formulation is.

A.4 Channel Pruning of Batch-Normalized CNN

We describe the basic principle and algorithm of our channel pruning technique.

A.4.1 Preliminaries

Pruning constant channels. Consider convolution with batch normalization:

$$x^{l+1} = \max\left\{\gamma^l \cdot \mathrm{BN}_{\mu^l,\sigma^l,\epsilon^l}(W^l * x^l) + \beta^l, 0\right\} \ .$$

For the ease of notation, we let $\gamma = \gamma^l$. Note that if some element in γ is set to zero, say, $\gamma[k] = 0$, its output image $x^{l+1}_{:,:,:,k}$ becomes a constant β_k, and a convolution of a constant image channel is almost everywhere constant (except for padding regions, an issue to be discussed later). Therefore, we show those constant image channels can be pruned while the same functionality of network is approximately kept:

- If the subsequent convolution layer does not have batch normalization,

  $$x^{l+2} = \max\left\{W^{l+1} * x^{l+1} + b^{l+1}, 0\right\} \ ,$$

  its values (a.k.a. elements in β) is absorbed into the bias term by the following equation

  $$b^{l+1}_{new} := b^{l+1} + I(\gamma = 0) \cdot \mathrm{ReLU}(\beta)^T \mathrm{sum\_reduced}(W^{l+1}_{:,:,:,\cdot}) \ ,$$

  such that

  $$x^{l+2} \approx \max\left\{W^{l+1} *_\gamma x^{l+1} + b^{l+1}_{new}, 0\right\} \ ,$$

where $*_\gamma$ denotes the convolution operator which is only calculated along channels indexed by non-zeros of γ. Remark that $W^* = \text{sum\_reduced}(W_{:,:,:,:})$ if $W^*_{a,b} = \sum_{i,j} W_{i,j,a,b}$.

- If the subsequent convolution layer has batch normalization,

$$x^{l+2} = \max\left\{\gamma^{l+1} \cdot \text{BN}_{\mu^{l+1},\sigma^{l+1},\epsilon^{l+1}}\left(W^{l+1} * x^{l+1}\right) + \beta^{l+1}, 0\right\} \ ,$$

instead its moving average is updated as

$$\mu^{l+1}_{new} := \mu^{l+1} - I(\gamma = 0) \cdot \text{ReLU}(\beta)^T \text{sum\_reduced}(W^{l+1}_{:,:,:,:}) \ ,$$

such that

$$x^{l+2} \approx \max\left\{\gamma^{l+1} \cdot \text{BN}_{\mu^{l+1}_{new},\sigma^{l+1},\epsilon^{l+1}}\left(W^{l+1} *_\gamma x^{l+1}\right) + \beta^{l+1}, 0\right\} \ .$$

Remark that the approximation ($\approx$) is strictly equivalence ($=$) if no padding is used in the convolution operator $*$, a feature that the parallel work [120] does not possess. When the original model uses padding in computing convolution layers, the network function is not strictly preserved after pruning. In our practice, we fine-tune the pruned network to fix such performance degradation at last. In short, we formulate the network pruning problem as simple as to set more elements in γ to zero. It is also much easier to deploy the pruned model, because no extra parameters or layers are introduced into the original model.

To better understand how it works in an entire CNN, imagine a channel-to-channel computation graph formed by the connections between layers. In this graph, each channel is a node, their inference dependencies are represented by directed edges. The γ parameter serves as a "dam" at each node, deciding whether let the received information "flood" through to other nodes following the graph. An end-to-end training of channel pruning is essentially like a flood control system. There suppose to be rich information of the input distribution, and in two ways, much of the original input information is lost along the way of CNN inference, and the useful part — that is supposed to be preserved by the network inference — should be label sensitive. Conventional CNN has one way to reduce information: transforming feature maps (non-invertible) via forward propagation. Our approach

introduces the other way: block information at each channel by forcing its output being constant using ISTA.

ISTA. Despite the gap between Lasso and sparsity in the non-convex settings, we found that ISTA [122] is still a useful sparse promoting method. But we just need to use it more carefully. Specifically, we adopt ISTA in the updates of γs. The basic idea is to project the parameter at every step of gradient descent to a potentially more sparse one subject to a proxy problem: let l denote the training loss of interest, at the $(t+1)$-th step, we set

$$\gamma_{t+1} = \min_{\gamma} \frac{1}{\mu_t} \|\gamma - \gamma_t + \mu_t \nabla_\gamma l_t\|^2 + \lambda \|\gamma\|_1 \ , \tag{23}$$

where $\nabla_\gamma l_t$ is the derivative with respect to γ computed at step t, μ_t is the learning rate, λ is the penalty. In the stochastic learning, $\nabla_\gamma l_t$ is estimated from a mini-batch at each step. Eq. (23) has closed form solution as

$$\gamma_{t+1} = \text{prox}_{\mu_t \lambda}(\gamma_t - \mu_t \nabla_\gamma l_t) \ ,$$

where $\text{prox}_\eta(x) = \max\{|x| - \eta, 0\} \cdot \text{sgn}(x)$. The ISTA method essentially serves as a "flood control system" in our end-to-end learning, where the functionality of each γ is like that of a dam. When γ is zero, the information flood is totally blocked, while $\gamma \neq 0$, the same amount of information is passed through in form of geometric quantities whose magnitudes are proportional to γ.

Scaling effect. One can also see that if γ is scaled by α meanwhile W^{l+1} is scaled by $1/\alpha$, that is,

$$\gamma := \alpha\gamma, \qquad W^{l+1} := \frac{1}{\alpha} W^{l+1}$$

the output x^{l+2} is unchanged for the same input x^l. Despite not changing the output, scaling of γ and W^{l+1} also scales the gradients $\nabla_\gamma l$ and $\nabla_{W^{l+1}} l$ by $1/\alpha$ and α, respectively. As we observed, the parameter dynamics of gradient learning with ISTA depends on the scaling factor α if one decides to choose it other than 1.0. Intuitively, if α is large, the optimization of W^{l+1} is progressed much slower than that of γ.

A.4.2 The Algorithm

We describe our algorithm below. The following method applies to both training from scratch or re-training from a pre-trained model. Given a training loss l, a convolutional neural net $\mathcal{N}$, and hyper-parameters ρ, α, μ_0, our method proceeds as follows:

1. **Computation of sparse penalty for each layer.** Compute the memory cost per channel for each layer denoted by λ^l and set the ISTA penalty for layer l to $\rho\lambda^l$. Here

$$\lambda^l = \frac{1}{I_w^i \cdot I_h^i} \left[k_w^l \cdot k_h^l \cdot c^{l-1} + \sum_{l' \in \mathcal{T}(l)} k_w^{l'} \cdot k_h^{l'} \cdot c^{l'} + I_w^l \cdot I_h^l \right] , \tag{24}$$

   where

   - $I_w^i \cdot I_h^i$ is the size of input image of the neural network.
   - $k_w^l \cdot k_h^l$ is the kernel size of the convolution at layer l. Likewise, $k_w^{l'} \cdot k_h^{l'}$ is the kernel size of subsequent convolution at layer l'.
   - $\mathcal{T}(l)$ represents the set of the subsequent convolutional layers of layer l
   - c^{l-1} denotes the channel size of the previous layer, which the l-th convolution operates over; and $c^{l'}$ denotes the channel size of one subsequent layer l'.
   - $I_w^l \cdot I_h^l$ is the image size of the feature map at layer l.

2. **γ-W rescaling trick.** For layers whose channels are going to get reduced, scale all γ^ls in batch normalizations by α meanwhile scale weights in their subsequent convolutions by $1/\alpha$.

3. **End-to-End training with ISTA on γ.** Train $\mathcal{N}$ by the regular SGD, with the exception that γ^ls are updated by ISTA, where the initial learning rate is μ_0. Train $\mathcal{N}$ until the loss l plateaus, the total sparsity of γ^ls converges, and Lasso $\rho \sum_l \lambda^l \|\gamma^l\|_1$ converges.

4. **Post-process to remove constant channels.** Prune channels in layer l whose elements in γ^l are zero and output the pruned model $\widetilde{\mathcal{N}}$ by absorbing all constant channels into subsequent layers (as described in the earlier section.).

5. γ-W **rescaling trick.** For γ^ls and weights in $\widetilde{\mathcal{N}}$ which were scaled in Step 2 before training, scale them by $1/\alpha$ and α respectively (scaling back).

6. Fine-tune $\widetilde{\mathcal{N}}$ using regular stochastic gradient learning.

Remark that choosing a proper α as used in Steps 2 and 5 is necessary for using a large $\mu_t \cdot \rho$ in ISTA, which makes the sparsification progress of γ^ls faster.

A.4.3 Guidelines for Tuning Hyper-parameters

We summarize the sensitivity of hyper-parameters and their impacts for optimization below:

- μ (learning rate): larger μ leads to fewer iterations for convergence and faster progress of sparsity. But if if μ too large, the SGD approach wouldn't converge.

- ρ (sparse penalty): larger ρ leads to more sparse model at convergence. If trained with a very large ρ, all channels will be eventually pruned.

- α (rescaling): we use α other than 1. only for pretrained models, we typically choose α from $\{0.001, 0.01, 0.1, 1\}$ and smaller α warms up the progress of sparsity.

We recommend the following parameter tuning strategy. First, check the cross-entropy loss and the regularization loss, select ρ such that these two quantities are comparable at the beginning. Second, choose a reasonable learning rate. Third, if the model is pretrained, check the average magnitude of γs in the network, choose α such that the magnitude of rescaled γ^l is around $100\mu\lambda^l\rho$. We found as long as one choose those parameters in the right range of magnitudes, the optimization progress is enough robust. Again one can monitor the mentioned three quantities during the training and terminate the iterations when all three quantities plateaus.

There are several patterns we found during experiments that may suggest the parameter tuning has not been successful. If during the first few epochs the Lasso-based regularization loss keeps decreasing linearly while the sparsity of γs stays near zero, one may decrease α and restart. If during the first few epochs the sparsity of γs quickly raise up to 100%, one may decrease ρ and restart. If during

the first few epochs the cross-entropy loss keeps at or increases dramatically to a non-informative level, one may decrease μ or ρ and restart.

A.5 Experiments

A.5.1 CIFAR-10 Experiment

We experiment with the standard image classification benchmark CIFAR-10 with two different network architectures: ConvNet and ResNet-20 [123]. We resize images to 32×32 and zero-pad them to 40×40. We pre-process the padded images by randomly cropping with size 32×32, randomly flipping, randomly adjusting brightness and contrast, and standardizing them such that their pixel values have zero mean and one variance.

ConvNet For reducing the channels in ConvNet, we are interested in studying whether one can easily convert a over-parameterized network into a compact one. We start with a standard 4-layer convolutional neural network whose network attributes are specified in Table 3. We use a fixed learning rate $\mu_t = 0.01$, scaling parameter $\alpha = 1.0$, and set batch size to 125.

Model A is trained from scratch using the base model with an initial warm-up $\rho = 0.0002$ for 30k steps, and then is trained by raising up ρ to 0.001. After the termination criterion are met, we prune the channels of the base model to generate a smaller network called model A. We evaluate the classification performance of model A with the running exponential average of its parameters. It is found that the test accuracy of model A is even better than the base model. Next, we start from the pre-trained model A to create model B by raising ρ up to 0.002. We end up with a smaller network called model B, which is about 1% worse than model A, but saves about one third parameters. Likewise, we start from the pre-trained model B to create model C. The detailed statistics and its pruned channel size are reported in Table 3. We also train a reference ConvNet from scratch whose channel sizes are 32-64-64-128 with totally 224,008 parameters and test accuracy being 86.3%. The referenced model is not as good as Model B, which has smaller number of parameters and higher accuracy.

We have two major observations from the experiment: (1) When the base network is over-parameterized, our approach not only significantly reduces the number

of channels of the base model but also improves its generalization performance on the test set. (2) Performance degradation seems unavoidable when the channels in a network are saturated, and our approach gives satisfactory trade-off between test accuracy and model efficiency.

			base	model A	model B	model C
layer	output	kernel	channel	channel	channel	channel
conv1	32×32	5×5	96	53	41	31
pool1	16×16	3×3				
conv2	16×16	5×5	192	86	64	52
pool2	8×8	3×3				
conv3	8×8	3×3	192	67	52	40
pool4	4×4	3×3				
fc	1×1	4×4	384	128	128	127
ρ				0.001	0.002	0.008
param. size			1,986,760	309,655	207,583	144,935
test accuracy (%)			89.0	89.5	87.6	86.0

Table 3. Comparisons between different pruned networks and the base network.

ResNet-20 We also want to verify our second observation with the state-of-art models. We choose the popular ResNet-20 as our base model for the CIFAR-10 benchmark, whose test accuracy is 92%. We focus on pruning the channels in the residual modules in ResNet-20, which has 9 convolutions in total. As detailed in Table 4, model A is trained from scratch using ResNet-20's network structure as its base model. We use a warm-up $\rho = 0.001$ for 30k steps and then train with $\rho = 0.005$. We are able to remove 37% parameters from ResNet-20 with only about 1 percent accuracy loss. Likewise, Model B is created from model A with a higher penalty $\rho = 0.01$.

A.5.2 ILSVRC2012 Experiment

We experiment our approach with the pre-trained ResNet-101 on ILSVRC2012 image classification dataset [123]. ResNet-101 is one of the state-of-the-art network architecture in ImageNet Challenge. We follow the standard pipeline to pre-process images to 224×224 for training ResNets. We adopt the pre-trained TensorFlow

	group - block	1-1	1-2	1-3	2-1	2-2	2-3	3-1	3-2	3-3
ResNet-20	channels	16	16	16	32	32	32	64	64	64
	param size.: 281,304									
	test accuracy (%): 92.0									
model A	channels	12	6	11	32	28	28	47	34	25
	param size.: 176,596									
	test accuracy (%): 90.9									
model B	channels	8	2	7	27	18	16	25	9	8
	param size.: 90,504									
	test accuracy (%): 88.8									

Table 4. Comparisons between ResNet-20 and its two pruned versions. The last columns are the number of channels of each residual modules after pruning.

ResNet-101 model whose single crop error rate is 23.6% with about 4.47×10^7 parameters.[2] We set the scaling parameter $\alpha = 0.01$, the initial learning rate $\mu_t = 0.001$, the sparsity penalty $\rho = 0.1$, and the batch size = 128 (across 4 GPUs). The learning rate is decayed every four epochs with rate 0.86. We create two pruned models from the different iterations of training ResNet-101: one has 2.36×10^7 parameters and the other has 1.73×10^7 parameters. We then fine-tune these two models using the standard way for training ResNet-101, and report their error rates. The Top-5 error rate increases of both models are less than 0.5%. The Top-1 error rates are summarized in Table 5. To our knowledge, only a few works have reported their performance on this very large-scale benchmark w.r.t. the Top-1 errors. We compare our approach with some recent works in terms of model parameter size, flops, and error rates. As shown in Table 5, our model v2 has achieved a compression ratio more than 2.5 while maintaining more than 1% lower error rates than that of other state-of-the-art models at comparable size of parameters.

In the first experiment (CIFAR-10), we train the network from scratch and allocate enough steps for both γ and W adjusting their own scales. Thus, initialization of an improper scale of γ-W is not really an issue given we optimize with enough steps. But for the pre-trained models which were originally optimized without any constraints of γ, the γ's scales are often unanticipated. It actually takes as many

[2] https://github.com/tensorflow/models/tree/master/slim

network	param size.	flops	error (%)	ratio
resnet-50 pruned [119]	$\sim 1.65 \times 10^7$	3.03×10^9	~ 26.8	66%
resnet-101 pruned (v2, ours)	1.73×10^7	3.69×10^9	**25.44**	39%
resnet-34 pruned [124]	1.93×10^7	2.76×10^9	27.8	89%
resnet-34	2.16×10^7	3.64×10^9	26.8	-
resnet-101 pruned (v1, ours)	2.36×10^7	4.47×10^9	**24.73**	53%
resnet-50	2.5×10^7	4.08×10^9	24.8	-
resnet-101	4.47×10^7	7.8×10^9	23.6	-

Table 5. Attributes of different versions of ResNet and their single crop errors on ILSVRC2012 benchmark. The last column means the parameter size of pruned model vs. the base model.

steps as that of training from scratch for γ to warm up. By adopting the rescaling trick setting α to a smaller value, we are able to skip the warm-up stage and quick start to sparsify γs. For example, it might take more than a hundred epoch to train ResNet-101, but it only takes about 5-10 epochs to complete the pruning and a few more epochs to fine-tune.

A.5.3 Image Foreground-Background Segmentation Experiment

As we have discussed about the two major observations in Section A.5.1, a more appealing scenario is to apply our approach in pruning channels of over-parameterized model. It often happens when one adopts a pre-trained network on a large task (such as ImageNet classification) and fine-tunes the model to a different and smaller task [118]. In this case, one might expect that some channels that have been useful in the first pre-training task are not quite contributing to the outputs of the second task.

We describe an image segmentation experiment whose neural network model is composed from an inception-like network branch and a densenet network branch. The entire network takes a 224×224 image and outputs binary mask at the same size. The inception branch is mainly used for locating the foreground objects while the densenet network branch is used to refine the boundaries around the segmented objects. This model was originally trained on multiple datasets.

In our experiment, we attempt to prune channels in both the inception branch and densenet branch. We set $\alpha = 0.01$, $\rho = 0.5$, $\mu_t = 2 \times 10^{-5}$, and batch size $= 24$.

We train the pre-trained base model until all termination criterion are met, and build the pruned model for fine-tuning. The pruned model saves 86% parameters and 81% flops of the base model. We also compare the fine-tuned pruned model with the pre-trained base model across different test benchmark. Mean IOU is used as the evaluation metric.[3] It shows that pruned model actually improves over the base model on four of the five test datasets with about $2\% \sim 5\%$, while it performs worse than the base model on the most challenged dataset DUT-Omron, whose foregrounds might contain multiple objects.

	base model	pruned model
test dataset (#images)	mIOU	mIOU
MSRA10K [125] (2,500)	83.4%	85.5%
DUT-Omron [126] (1,292)	83.2%	79.1%
Adobe Flickr-portrait [127] (150)	88.6%	93.3%
Adobe Flickr-hp [127] (300)	84.5%	89.5%
COCO-person [128] (50)	84.1%	87.5%
param. size	1.02×10^7	1.41×10^6
flops	5.68×10^9	1.08×10^9

Table 6. mIOU reported on different test datasets for the base model and the pruned model.

A.6 Conclusions

We proposed a model pruning technique that focuses on simplifying the computation graph of a deep convolutional neural network. Our approach adopts ISTA to update the γ parameter in batch normalization operator embedded in each convolution. To accelerate the progress of model pruning, we use a γ-W rescaling trick before and after stochastic training. Our method cleverly avoids some possible numerical difficulties such as mentioned in other regularization-based related work, hence is easier to apply for practitioners. We empirically validated our method through several benchmarks and showed its usefulness and competitiveness in building compact CNN models.

[3] https://www.tensorflow.org/api_docs/python/tf/metrics/mean_iou

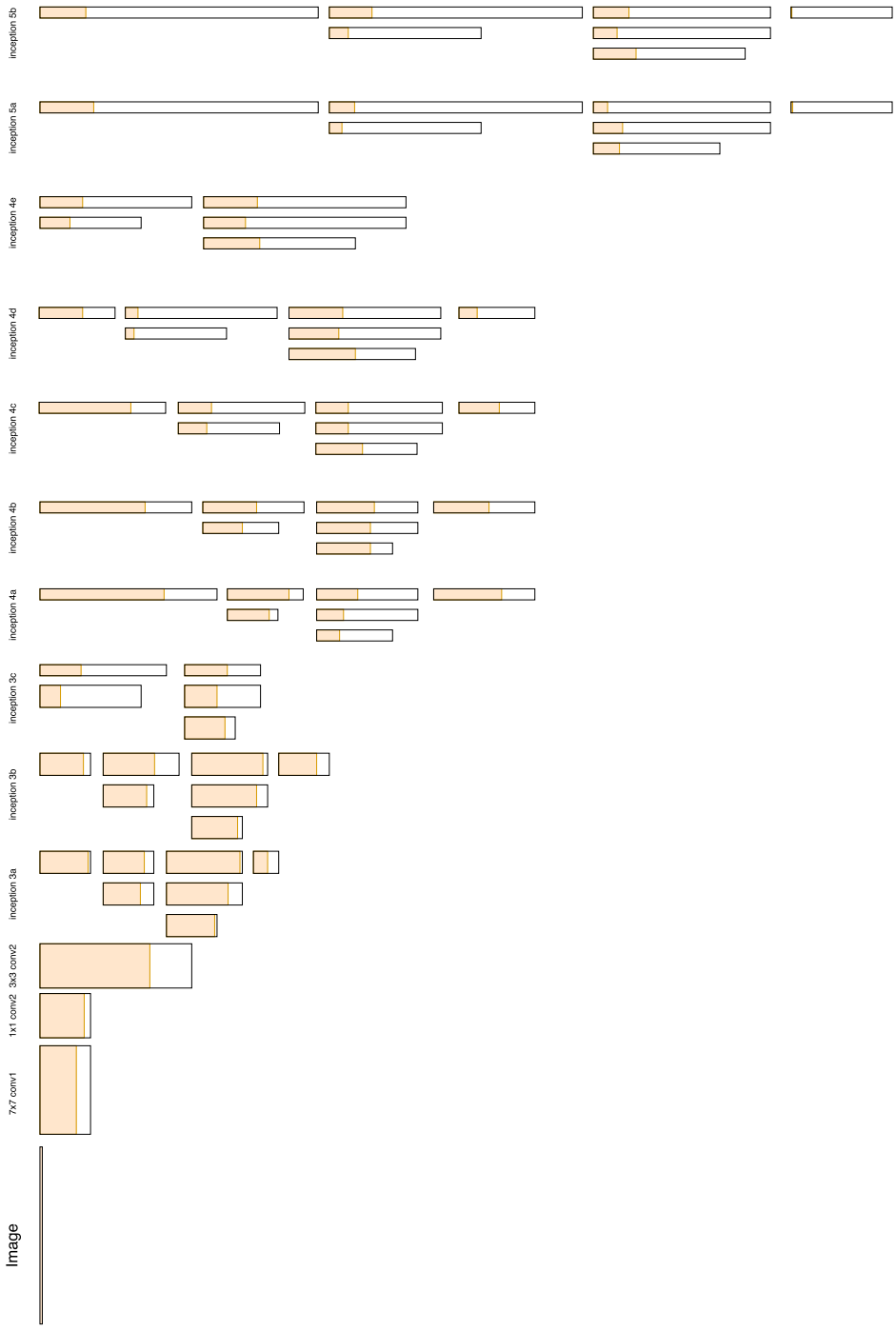


Figure 12. Visualization of the number of pruned channels at each convolution in the inception branch. Colored regions represents the number of channels kept. The height of each bar represents the size of feature map, and the width of each bar represents the size of channels. It is observed that most of channels in the bottom layers are kept while most of channels in the top layers are pruned.

Bibliography

[1] Villani, C. (2003) *Topics in Optimal Transportation*, 58, American Mathematical Soc.

[2] Cuturi, M. and A. Doucet (2014) "Fast computation of Wasserstein barycenters," in *Proceedings of International Conference on Machine Learning*, pp. 685–693.

[3] Cuturi, M. and G. Peyré (2016) "A smoothed dual approach for variational Wasserstein problems," *SIAM Journal on Imaging Sciences*, **9**(1), pp. 320–343.

[4] Zhou, D., J. Li, and H. Zha (2005) "A new mallows distance based metric for comparing clusterings," in *Proceedings of International Conference on Machine Learning*, ACM, pp. 1028–1035.

[5] Kusner, M., Y. Sun, N. Kolkin, and K. Weinberger (2015) "From word embeddings to document distances," in *Proceedings of International Conference on Machine Learning*, pp. 957–966.

[6] Vinh, N. X., J. Epps, and J. Bailey (2010) "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *Journal of Machine Learning Research (JMLR)*, **11**, pp. 2837–2854.

[7] Ye, J., J. Li, M. G. Newman, R. B. Adams, and J. Z. Wang (2017) "Probabilistic multigraph modeling for improving the quality of crowdsourced affective data," *IEEE Transactions on Affective Computing*.

[8] Ye, J., P. Wu, J. Z. Wang, and J. Li (2017) "Fast discrete distribution clustering using Wasserstein barycenter with sparse support," *IEEE Transactions on Signal Processing*, **65**(9), pp. 2317–2332.

[9] Ye, J., Y. Li, Z. Wu, J. Z. Wang, W. Li, and J. Li (2017) "Determining gains acquired from word embedding quantitatively using discrete distribution Clustering," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 1847–1856.

[10] Ye, J., J. Z. Wang, and J. Li (2017) "A simulated annealing based inexact oracle for Wasserstein loss minimization," in *Proceedings of International Conference on Machine Learning*, pp. 3940–3948.

[11] Ye, J. and J. Li (2014) "Scaling up discrete distribution clustering using ADMM," in *Image Processing (ICIP), 2014 IEEE International Conference on*, IEEE, pp. 5267–5271.

[12] Chen, Y., J. Ye, and J. Li (2016) "A distance for HMMs based on aggregated Wasserstein metric and state registration," in *Proceedings of European Conference on Computer Vision*, Springer, pp. 451–466.

[13] Lin, L., J. Ye, and J. Li (2017) "Label switching problem in Gaussian mixture models: an approach based on Wasserstein barycenters," *Biometrika (under review)*.

[14] Li, J. and F. Zhang (2018) "Geometry-Sensitive ensemble mean based on Wasserstein barycenters: proof-of-concept on cloud simulations," *Journal of Computational and Graphical Statistics (accepted)*.

[15] Peleg, S., M. Werman, and H. Rom (1989) "A unified approach to the change of resolution: Space and gray-level," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **11**(7), pp. 739–742.

[16] Rubner, Y., C. Tomasi, and L. J. Guibas (2000) "The Earth Mover's distance as a metric for image retrieval," *International Journal on Computer Vision*, **40**(2), pp. 99–121.

[17] Logan, B. and A. Salomon (2001) "A music similarity function based on signal analysis," in *Proceedings of International Conference on Multimedia and Expo (ICME)*, IEEE, pp. 745–748.

[18] Grauman, K. and T. Darrell (2005) "The pyramid match kernel: Discriminative classification with sets of image features," in *Proceedings of International Conference on Computer Vision (ICCV)*, vol. 2, IEEE, pp. 1458–1465.

[19] Li, J. and J. Z. Wang (2008) "Real-time computerized annotation of pictures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**(6), pp. 985–1002.

[20] Agueh, M. and G. Carlier (2011) "Barycenters in the Wasserstein space," *SIAM Journal on Mathematical Analysis*, **43**(2), pp. 904–924.

[21] Benamou, J.-D., G. Carlier, M. Cuturi, L. Nenna, and G. Peyré (2015) "Iterative Bregman projections for regularized transportation problems," *SIAM Journal on Scientific Computing*, **37**(2), pp. A1111–A1138.

[22] Seguy, V. and M. Cuturi (2015) "Principal geodesic analysis for probability measures under the optimal transport metric," in *Advances in Neural Information Processing Systems*, pp. 3294–3302.

[23] Rolet, A., M. Cuturi, and G. Peyré (2016) "Fast dictionary learning with a smoothed Wasserstein loss," in *Proceedings of International Conference on Artificial Intelligence and Statistics*.

[24] Sandler, R. and M. Lindenbaum (2009) "Nonnegative matrix factorization with Earth Mover's distance metric," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 1873–1880.

[25] Bonneel, N., G. Peyré, and M. Cuturi (2016) "Wasserstein barycentric coordinates: histogram regression using optimal transport," *ACM Transactions on Graphics*, **35**(4).

[26] Frogner, C., C. Zhang, H. Mobahi, M. Araya, and T. A. Poggio (2015) "Learning with a Wasserstein loss," in *Advances in Neural Information Processing Systems*, pp. 2044–2052.

[27] Monge, G. (1781) *Mémoire sur la théorie des déblais et des remblais*, De l'Imprimerie Royale.

[28] Cuturi, M. (2013) "Sinkhorn distances: Lightspeed computation of optimal transport," in *Advances in Neural Information Processing Systems*, pp. 2292–2300.

[29] Kalantari, B., I. Lari, F. Ricca, and B. Simeone (2008) "On the complexity of general matrix scaling and entropy minimization via the RAS algorithm," *Mathematical Programming*, **112**(2), pp. 371–401.

[30] Altschuler, J., J. Weed, and P. Rigollet (2017) "Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration," in *Advances in Neural Information Processing Systems*, pp. 1961–1971.

[31] Wang, H. and A. Banerjee (2014) "Bregman alternating direction method of multipliers," in *Advances in Neural Information Processing Systems*, pp. 2816–2824.

[32] Ye, J. (2017), "New numerical tools for optimal transport and their machine learning applications," .

[33] Orlin, J. B. (1993) "A faster strongly polynomial minimum cost flow algorithm," *Operations Research*, **41**(2), pp. 338–350.

[34] Kirkpatrick, S., J. Gelatt, C. Daniel, and M. P. Vecchi (1983) "Optimization by simulated annealing," *Science*, **220**(4598), pp. 671–680.

[35] Corana, A., M. Marchesi, C. Martini, and S. Ridella (1987) "Minimizing multi-modal functions of continuous variables with the "simulated annealing" algorithm," *ACM Transactions on Mathematical Software*, **13**(3), pp. 262–280.

[36] Benamou, J.-D., G. Carlier, and L. Nenna (2016) "A numerical method to solve multi-marginal optimal transport problems with Coulomb cost," in *Splitting Methods in Communication, Imaging, Science, and Engineering*, Springer, pp. 577–601.

[37] Banerjee, A., S. Merugu, I. S. Dhillon, and J. Ghosh (2005) "Clustering with Bregman divergences," *Journal of Machine Learning Research*, **6**, pp. 1705–1749.

[38] Zhang, Y., J. Z. Wang, and J. Li (2015) "Parallel massive clustering of discrete distributions," *ACM Transactions on Multimedia Computing, Communications and Applications*, **11**(4), pp. 49:1–49:24.

[39] Pele, O. and M. Werman (2009) "Fast and robust Earth Mover's distances," in *Proceedings of International Conference on Computer Vision*, IEEE, pp. 460–467.

[40] Anderes, E., S. Borgwardt, and J. Miller (2015) "Discrete Wasserstein Barycenters: Optimal Transport for Discrete Data," *arXiv preprint arXiv:1507.07218*.

[41] Carlier, G., A. Oberman, and E. Oudet (2014) "Numerical methods for matching for teams and Wasserstein barycenters," *arXiv preprint arXiv:1411.3602*.

[42] Solomon, J., F. de Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas (2015) "Convolutional Wasserstein distances: efficient optimal transportation on geometric domains," *ACM Transactions on Graphics*, **34**(4), pp. 66:1–66:11.

[43] Rabin, J., G. Peyré, J. Delon, and M. Bernot (2011) "Wasserstein barycenter and its application to texture mixing," in *Scale Space and Variational Methods in Computer Vision*, Springer, pp. 435–446.

[44] Boyd, S., N. Parikh, E. Chu, B. Peleato, and J. Eckstein (2011) "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, **3**(1), pp. 1–122.

[45] Tang, Y., L. H. U, Y. Cai, N. Mamoulis, and R. Cheng (2013) "Earth Mover's Distance Based Similarity Search at Scale," *Proceedings of the VLDB Endowment*, **7**(4), pp. 313–324.

[46] Bregman, L. M. (1967) "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Computational Mathematics and Mathematical Physics*, **7**(3), pp. 200–217.

[47] Censor, Y. and S. A. Zenios (1992) "Proximal minimization algorithm with D-functions," *Journal of Optimization Theory and Applications*, **73**(3), pp. 451–464.

[48] Eckstein, J. (1993) "Nonlinear proximal point algorithms using Bregman functions, with applications to convex programming," *Mathematics of Operations Research*, **18**(1), pp. 202–226.

[49] Srivastava, A., I. Jermyn, and S. Joshi (2007) "Riemannian analysis of probability density functions with applications in vision," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 1–8.

[50] Elkan, C. (2003) "Using the triangle inequality to accelerate k-means," in *Proceedings on International Conference on Machine Learning*, vol. 3, pp. 147–153.

[51] Greene, D. and P. Cunningham (2006) "Practical solutions to the problem of diagonal dominance in kernel document clustering," in *Proceedings of International Conference on Machine Learning*, ACM, pp. 377–384.

[52] Wu, Z., C. Liang, and C. L. Giles (2015) "Storybase: Towards building a knowledge base for news events," in *ACL-IJCNLP 2015*, pp. 133–138.

[53] Rosenberg, A. and J. Hirschberg (2007) "V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure." in *Proceedings of EMNLP-CoNLL*, vol. 7, pp. 410–420.

[54] Arthur, D. and S. Vassilvitskii (2007) "k-means++: The advantages of careful seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, Society for Industrial and Applied Mathematics, pp. 1027–1035.

[55] Globerson, A. and S. Roweis (2006) "Nightmare at test time: robust learning by feature deletion," in *Proceedings of International conference on Machine learning*, ACM, pp. 353–360.

[56] Wan, X. (2007) "A novel document similarity measure based on Earth MoverâĂŹs distance," *Information Sciences*, **177**(18), pp. 3718–3730.

[57] Pennington, J., R. Socher, and C. D. Manning (2014) "Glove: Global vectors for word representation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, ACL, pp. 1532–1543.

[58] Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013) "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 3111–3119.

[59] Rand, W. M. (1971) "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, **66**(336), pp. 846–850.

[60] Hubert, L. and P. Arabie (1985) "Comparing partitions," *Journal of Classification*, **2**(1), pp. 193–218.

[61] Hoffman, M., F. R. Bach, and D. M. Blei (2010) "Online learning for latent dirichlet allocation," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 856–864.

[62] Nemirovski, A. (2004) "Prox-method with rate of convergence O (1/t) for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems," *SIAM Journal on Optimization*, **15**(1), pp. 229–251.

[63] Beck, A. and M. Teboulle (2003) "Mirror descent and nonlinear projected subgradient methods for convex optimization," *Operations Research Letters*, **31**(3), pp. 167–175.

[64] Bubeck, S. et al. (2015) "Convex optimization: Algorithms and complexity," *Foundations and Trends® in Machine Learning*, **8**(3-4), pp. 231–357.

[65] Chizat, L., G. Peyré, B. Schmitzer, and F.-X. Vialard (2016) "Scaling algorithms for unbalanced transport problems," *arXiv preprint arXiv:1607.05816*.

[66] Schmitzer, B. (2016) "Stabilized Sparse Scaling Algorithms for Entropy Regularized Transport Problems," *arXiv preprint arXiv:1610.06519*.

[67] Krichene, W., A. Bayen, and P. L. Bartlett (2015) "Accelerated mirror descent in continuous and discrete time," in *Advances in Neural Information Processing Systems*, pp. 2827–2835.

[68] Mikolov, T., W.-t. Yih, and G. Zweig (2013) "Linguistic regularities in continuous space word representations." in *HLT-NAACL*, pp. 746–751.

[69] Singh, S., A. Subramanya, F. Pereira, and A. McCallum (2011) "Large-scale cross-document coreference using distributed inference and hierarchical models," in *Proceedings of ACL-HLT*, Association for Computational Linguistics, pp. 793–803.

[70] Zhai, Z., B. Liu, H. Xu, and P. Jia (2011) "Clustering product features for opinion mining," in *Proceedings of International Conference on Web Search and Data Mining (WSDM)*, ACM, pp. 347–354.

[71] Strehl, A. and J. Ghosh (2003) "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research (JMLR)*, **3**, pp. 583–617.

[72] Sparck Jones, K. (1972) "A statistical interpretation of term specificity and its application in retrieval," *J. Documentation*, **28**(1), pp. 11–21.

[73] Belkin, M. and P. Niyogi (2001) "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering." in *Advances in Neural Information Processing Systems (NIPS)*, vol. 14, pp. 585–591.

[74] Deerwester, S. C., S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman (1990) "Indexing by latent semantic analysis," *J. American Soc. Information Science*, **41**(6), pp. 391–407.

[75] He, X. and P. Niyogi (2004) "Locality preserving projections," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 16, MIT, p. 153.

[76] Cai, D., X. He, and J. Han (2005) "Document clustering using locality preserving indexing," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, **17**(12), pp. 1624–1637.

[77] Lee, D. D. and H. S. Seung (1999) "Learning the parts of objects by non-negative matrix factorization," *Nature*, **401**(6755), pp. 788–791.

[78] Xu, W., X. Liu, and Y. Gong (2003) "Document clustering based on non-negative matrix factorization," in *ACM SIGIR Conference on Research and Development in Informaion Retrieval*, ACM, pp. 267–273.

[79] Blei, D. M., A. Y. Ng, and M. I. Jordan (2003) "Latent dirichlet allocation," *Journal of Machine Learning Research*, **3**, pp. 993–1022.

[80] Le, Q. and T. Mikolov (2014) "Distributed representations of sentences and documents," in *Proceedings of International Conference on Machine Learning*, pp. 1188–1196.

[81] Huang, G., C. Guo, M. J. Kusner, Y. Sun, F. Sha, and K. Q. Weinberger (2016) "Supervised word mover's distance," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 4862–4870.

[82] Cuturi, M. and D. Avis (2014) "Ground metric learning," *Journal of Machine Learning Research*, **15**(1), pp. 533–564.

[83] Datta, R., D. Joshi, J. Li, and J. Z. Wang (2006) "Studying aesthetics in photographic images using a computational approach," in *Proceedings of European Conference on Computer Vision*, Springer, pp. 288–301.

[84] Lu, X., P. Suryanarayan, R. B. Adams Jr, J. Li, M. G. Newman, and J. Z. Wang (2012) "On shape and the computability of emotions," in *Proceedings of the 20th ACM International Conference on Multimedia*, ACM, pp. 229–238.

[85] Lu, X. (2015) *Visual Characteristics for Computational Prediction of Aesthetics and Evoked Emotions*, Ph.D. thesis, The Pennsylvania State University, chapter 5.
URL https://etda.libraries.psu.edu/catalog/28857

[86] Dawid, A. P. and A. M. Skene (1979) "Maximum likelihood estimation of observer error-rates using the EM algorithm," *Applied Statistics*, pp. 20–28.

[87] Hui, S. L. and S. D. Walter (1980) "Estimating the error rates of diagnostic tests," *Biometrics*, pp. 167–171.

[88] Smyth, P., U. M. Fayyad, M. C. Burl, P. Perona, and P. Baldi (1995) "Inferring ground truth from subjective labeling of Venus images," in *Advances in Neural Information Processing Systems*, pp. 1085–1092.

[89] Demartini, G., D. E. Difallah, and P. Cudré-Mauroux (2012) "ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking," in *Proceedings of the 21st International Conference on World Wide Web*, ACM, pp. 469–478.

[90] Raykar, V. C., S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy (2010) "Learning from crowds," *Journal of Machine Learning Research*, **11**, pp. 1297–1322.

[91] Liu, Q., J. Peng, and A. T. Ihler (2012) "Variational inference for crowdsourcing," in *Advances in Neural Information Processing Systems*, pp. 692–700.

[92] Raykar, V. C. and S. Yu (2012) "Eliminating spammers and ranking annotators for crowdsourced labeling tasks," *Journal of Machine Learning Research*, **13**(1), pp. 491–518.

[93] Whitehill, J., T.-f. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo (2009) "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *Advances in Neural Information Processing Systems*, pp. 2035–2043.

[94] Sheshadri, A. and M. Lease (2013) "SQUARE: A benchmark for research on computing crowd consensus," in *First AAAI Conference on Human Computation and Crowdsourcing*, pp. 156–164.

[95] Wang, Y. J. and G. Y. Wong (1987) "Stochastic blockmodels for directed graphs," *Journal of the American Statistical Association*, **82**(397), pp. 8–19.

[96] Nowicki, K. and T. A. B. Snijders (2001) "Estimation and prediction for stochastic blockstructures," *Journal of the American Statistical Association*, **96**(455), pp. 1077–1087.

[97] Hoff, P. D., A. E. Raftery, and M. S. Handcock (2002) "Latent space approaches to social network analysis," *Journal of the American Statistical Association*, **97**(460), pp. 1090–1098.

[98] Airoldi, E. M., D. M. Blei, S. E. Fienberg, and E. P. Xing (2009) "Mixed membership stochastic blockmodels," in *Advances in Neural Information Processing Systems*, pp. 33–40.

[99] Kim, M. and J. Leskovec (2012) "Latent Multi-group Membership Graph Model," in *Proceedings of International Conference on Machine Learning*, pp. 1719–1726.

[100] Kemp, C., J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda (2006) "Learning systems of concepts with an infinite relational model," in *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, pp. 381–388.

[101] Kemp, C. and J. B. Tenenbaum (2008) "The discovery of structural form," *Proceedings of the National Academy of Sciences*, **105**(31), pp. 10687–10692.

[102] Jordan, M. I., Z. Ghahramani, T. S. Jaakkola, and L. K. Saul (1999) "An introduction to variational methods for graphical models," *Machine Learning*, **37**(2), pp. 183–233.

[103] Bernardo, J., M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, M. West, et al. (2003) "The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures," *Bayesian Statistics*, **7**, pp. 453–464.

[104] Johnson, N. L., S. Kotz, and N. Balakrishnan (1995) "Chapter 21: beta distributions," *Continuous Univariate Distributions Vol. 2*.

[105] Gentner, D. (2010) "Bootstrapping the mind: Analogical processes and symbol systems," *Cognitive Science*, **34**(5), pp. 752–775.

[106] Gentner, D. and S. Christie (2010) "Mutual bootstrapping between language and analogical processing," *Language and Cognition*, **2**(2), pp. 261–283.

[107] He, Y., X. Zhang, and J. Sun (2017) "Channel pruning for accelerating very deep neural networks," in *Proceedings of International Conference on Computer Vision*.

[108] Zhang, X., J. Zou, K. He, and J. Sun (2016) "Accelerating very deep convolutional networks for classification and detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38**(10), pp. 1943–1955.

[109] Han, S., J. Pool, J. Tran, and W. Dally (2015) "Learning both weights and connections for efficient neural network," in *Advances in Neural Information Processing Systems*, pp. 1135–1143.

[110] Anwar, S., K. Hwang, and W. Sung (2017) "Structured pruning of deep convolutional neural networks," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, **13**(3), p. 32.

[111] Ioffe, S. and C. Szegedy (2015) "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of International Conference on Machine Learning*, pp. 448–456.

[112] LeCun, Y., J. S. Denker, and S. A. Solla (1990) "Optimal brain damage," in *Advances in Neural Information Processing Systems*, pp. 598–605.

[113] Hassibi, B. and D. G. Stork (1993) "Second order derivatives for network pruning: Optimal brain surgeon," in *Advances in Neural Information Processing Systems*, pp. 164–171.

[114] Wen, W., C. Wu, Y. Wang, Y. Chen, and H. Li (2016) "Learning structured sparsity in deep neural networks," in *Advances in Neural Information Processing Systems*, pp. 2074–2082.

[115] Zhou, H., J. M. Alvarez, and F. Porikli (2016) "Less is more: Towards compact CNNs," in *Proceedings of European Conference on Computer Vision*, Springer, pp. 662–677.

[116] Alvarez, J. M. and M. Salzmann (2016) "Learning the number of neurons in deep networks," in *Advances in Neural Information Processing Systems*, pp. 2270–2278.

[117] Lebedev, V. and V. Lempitsky (2016) "Fast convnets using group-wise brain damage," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2554–2564.

[118] Molchanov, P., S. Tyree, T. Karras, T. Aila, and J. Kautz (2017) "Pruning convolutional neural networks for resource efficient transfer learning," in *Proceedings of International Conference on Learning Representations*.

[119] Huang, Z. and N. Wang (2017) "Data-Driven Sparse Structure Selection for Deep Neural Networks," *arXiv preprint arXiv:1707.01213*.

[120] Liu, Z., J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang (2017) "Learning Efficient Convolutional Networks through Network Slimming," *arXiv preprint arXiv:1708.06519*.

[121] Dinh, L., R. Pascanu, S. Bengio, and Y. Bengio (2017) "Sharp minima can generalize for deep nets," in *Proceedings of International Conference on Machine Learning*.

[122] Beck, A. and M. Teboulle (2009) "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, **2**(1), pp. 183–202.

[123] He, K., X. Zhang, S. Ren, and J. Sun (2016) "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.

[124] Li, H., A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf (2017) "Pruning filters for efficient convnets," in *Proceedings of International Conference on Learning Representations*.

[125] Liu, T., Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum (2011) "Learning to detect a salient object," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33**(2), pp. 353–367.

[126] Yang, C., L. Zhang, H. Lu, X. Ruan, and M.-H. Yang (2013) "Saliency detection via graph-based manifold ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3166–3173.

[127] Shen, X., A. Hertzmann, J. Jia, S. Paris, B. Price, E. Shechtman, and I. Sachs (2016) "Automatic portrait segmentation for image stylization," in *Computer Graphics Forum*, vol. 35, Wiley Online Library, pp. 93–102.

[128] Lin, T.-Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick (2014) "Microsoft coco: Common objects in context," in *Proceedings of European Conference on Computer Vision*, Springer, pp. 740–755.

Vita

Jianbo Ye

Jianbo Ye was born in Zhejiang, China on August 21, 1989. He received the B.Sc. degree in Mathematics from University of Science and Technology of China in 2011. He was a research postgraduate at the Department of Computer Science, The University of Hong Kong from 2011 to 2012. As a PhD student at College of Information Sciences and Technology of Penn State, Ye was working with James Z. Wang and Jia Li on machine learning, optimization methods and computational statistics, as well as applications in emotion modeling, computational linguistics, and meteorology big data. He has published three first-authored papers in high-impact journals and about ten papers in peer-reviewed conference proceedings. In addition, he has filed four US patents in areas of artificial intelligence and computer graphics. He has worked as a research intern at Intel Corp. and Adobe Systems Inc., in 2013 and 2017, respectively. His thesis research has focused on developing scalable and robust numerical algorithms that apply optimal transport theory and Wasserstein geometry to machine learning models and justifying their real-world impacts in terms of data analytics and scalability. During 2017 and 2018, Ye was invited to present his thesis research at high-impact conferences (e.g. CMO-BIRS Workshop — a high-profile mathematician workshop, SIAM Southeastern Atlantic Sectional Conference) and research institutes (Carnegie Mellon University, Toutiao AI Labs). He was a Kaggle gold and silver medalist and an author of neural network library in Scala. His general research interests include optimal transport in machine learning, deep learning, and affective computing. After graduation in May 2018, Ye will join Amazon Lab126 as an Applied Scientist.