University of Alberta

Bregman Divergence Clustering: A Convex Approach

by

Hao Cheng

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

Fall 2013
Edmonton, Alberta

Abstract

Due to its wide application in various fields, clustering, as a fundamental unsupervised learning problem, has been intensively investigated over the past few decades. Unfortunately, standard clustering formulations are known to be computationally intractable. Although many convex relaxations of clustering have been proposed to overcome the challenge of computational intractability, current formulations of clustering remain largely restricted to spherical Gaussian or discriminative models and are susceptible to imbalanced clusters. To address these shortcomings, we propose a new class of convex relaxations that can be flexibly applied to more general forms of Bregman divergence clustering. By basing these new formulations on *normalized equivalence relation matrix*, we retain additional control on relaxation quality, which allows improvement in clustering quality. We furthermore develop optimization methods that improve scalability by exploiting recent implicit matrix norm methods. We find that the new formulations are able to efficiently produce tighter clusterings that improve the accuracy of state of the art methods.

Acknowledgements

First and foremost, I would like to give my sincere thanks to my two great supervisors Dale Schuurmans and Csaba Szepesvári for their knowledgeable advice, kindness and flexibility in allowing me to explore a variety of topics of my own interest. Both the formal courses and discussions with them always give me additional insight, knowledge and new perspective which constantly motivate me on my way of exploring the machine learning research problems. I also would like to thank my committee chair Pierre Boulanger for his constructive feedback.

During my two years here, I've gotten the opportunities to work and collaborate on various interesting projects. I would like to thank Xinhua Zhang for his time , advice and generous support for our project on Bregman divergence clustering. I also want to thank Yaoliang Yu for his support and collaboration which inspires me to explore the statistical and algorithmic side of machine learning.

I am really grateful for all the great friends and colleagues I have met here in Edmonton. You have made this experience extremely worthwhile.

Finally, I would like to thank my grandparents for their continuous love and support.

Contents

List of Tables

List of Symbols

Symbol	Description	First use
$\mathbf{X}$	observed variable	4
$\mathbf{Y}$	latent class variable	4
$P(\mathbf{X}\|\mathbf{Y})$	generative conditional likelihood	4
$P(\mathbf{X}, \mathbf{Y})$	joint likelihood	5
$P(\mathbf{X})$	marginal likelihood	5
$P(\mathbf{Y})$	marginal likelihood	5
$P(\mathbf{Y}\|\mathbf{X})$	discriminative conditional likelihood	6
F	potential function for defining a Bregman divergence	6
f	transfer function	6
$d_F(\cdot, \cdot)$	Bregman divergence defined by potential function F	6
$D_F(\cdot, \cdot)$	Bregman divergence in the matrix form	8
F^*	Fenchel conjugate of F	6
$p(\cdot)$	probability density function	8
$\boldsymbol{\theta}$	natural parameters	8
$\phi(\cdot)$	sufficient statistics	8
$Z(\boldsymbol{\theta})$	partition function	8
$A(\boldsymbol{\theta})$	log-parition function	8
$\dagger$	pseudo inverse	16
$\mathrm{diag}(\cdot)$	diagonals of a matrix	16
$\mathbf{1}$	all one vector	16
Δ_d	d-dimensional simplex	17

Chapter 1

Introduction

Discovering latent class structure in data, *i.e. clustering*, is a fundamental problem in many fields, such as bioformatics, machine learning, and statistics. Given data, the task is to assign each observation a latent cluster label or distribution over cluster labels based on some notion of similarity. Because of its important role in exploratory data analysis, clustering has a wide range of applications, including:

- *Astronomy*: A catalog of billions of sky objects represented by their radiation in frequency bands can be clustered into similar objects, *e.g.*, galaxies, nearby stars, quasars.
- *Biology*: DNA sequences can be clustered based on edit distance.
- *Marketing*: Customers can be clustered based on their profile as well as their purchase histories or products can be clustered based on the sets of customers.
- *Banking*: Credit card behavior (*i.e.*fraud vs normal use) is clustered based on cardholder's transaction history.
- *WWW*: Documents can be clustered based on similar words.

A common goal of clustering formulations is to promote intra-cluster similarity and inter-cluster dissimilarity. However, there is no best clustering criteria. In practice, users have to supply various forms of prior knowledge. In addition to specifying prior information on the number of clusters, some applications require a strict partition while others require a probabilistic assignment. Also, some applications might only require finding cluster representatives while others might require discovering useful unknown properties from the data.

Clustering has a long history, with diverse approaches proposed. Unfortunately, computational tractability remains a fundamental challenge: standard clustering formulations are *NP*-hard (Aloise et al., 2009; Dasgupta, 2008; Arora & Kannan, 2005) and additional problem structure must be postulated before efficient solutions can be guaranteed. Meanwhile, standard clustering formulations are also efficiently approximable (Kumar et al., 2004; Arthur & Vassilvitskii, 2007), and much work has sought practical algorithms that improve solution quality, even in lieu of theoretical bounds. A popular approach for approximation is through convex relaxation that can be solved in polynomial time. Therefore, in this thesis, I investigate possible convex relaxations for common clustering paradigms with corresponding efficient optimization algorithms.

1.1 Contributions

The main contributions of this thesis are:

1. For centroid-based Bregman divergence clustering, we develop a new family of convex relaxations that use a *normalized* equivalence relation matrix to improve the quality of previous convex relaxations. We also analyze the tightness of this new convex relaxation.

2. Based on the analysis of *normalized* equivalence relations, we design an induced matrix norm technique that can be applied across a broad range of convex relaxations, which results in efficient optimization algorithms for the corresponding nonlinear semidefinite programs (SDPs).

3. Finally, by using a standard rounding procedure, we observe that the resulting clustering algorithms provide superior or comparable empirical performance to current approaches on various kinds of datasets. In particular, our formulation of discriminative clustering is at least 10 times faster than existing approaches, while automatically alleviating the problem of imbalanced cluster assignment.

1.2 Organization

In this thesis, I will first review related work on clustering in Chapter 2, then present background on the general loss models I consider (Bregman divergences) and the underlying optimization strategy I will primarily use (generalized conditional gradient method). Then, I will present a new family of convex relaxations with efficient algorithms for hard conditional clustering, discriminative

clustering, and hard joint clustering, respectively, in Chapters 3 to 6. Corresponding experimental evaluations of the proposed convex relaxations will be presented in each chapter. Finally, the conclusions and potential future work will be discussed in Chapter 6. The results of this thesis have been published in (Cheng et al., 2013).

Chapter 2

Background

In this chapter, I will provide the necessary background on clustering formulations, Bregman divergences and the generalized conditional gradient method for optimization.

2.1 General Formulations of Clustering

Two of the most important paradigms for centroid-based clustering are based on *generative* versus *discriminative* modeling, with generative clustering consisting of hard clustering with conditional models, hard clustering with joint models, and soft clustering with joint models. In hard clustering, one seeks a disjoint partition of data points such that each data point belongs to just one cluster. In soft clustering, each observation is assigned a certain probability of being a member to each of the clusters.

Traditionally, clustering has used *generative models* to capture interesting latent structure in data. Let $\mathbf{X}$ denote the observed variable and $\mathbf{Y}$ denote a latent class variable. The simplest generative approach optimizes the conditional model $P(\mathbf{X}|\mathbf{Y})$ only, with $\mathbf{Y}$ assigned the most likely value; this is also known as *hard conditional* clustering. When $P(\mathbf{X}|\mathbf{Y})$ is Gaussian, a popular approach is to use the hard k-means clustering algorithm (MacQueen, 1967) where one alternates between optimizing $\mathbf{Y}$ and the conditional model. Banerjee et al. (2005) extended the k-means formulation to general exponential family by modeling $P(\mathbf{X}|\mathbf{Y})$ with Bregman divergences.

Although hard conditional clustering provides a standard baseline, finding global solutions in this case is intractable; efficient methods are only known when the number of clusters or the dimensionality of the space is constrained (Hansen et al., 1998; Inaba et al., 1994). Consequently, there has been significant work on developing approximations, particularly via convex relaxations

that can be solved in polynomial time. For example, Zha et al. (2001) derived a convex quadratic reformulation of conditional Gaussian clustering, and Peng & Wei (2007) obtained a tighter semi-definite programming (SDP) relaxation. By analyzing the complete positivity (CP) properties of the resulting constraint, Zass & Shashua (2005) propose an approximation for Gaussian clustering based on CP factorization. These can be further extended to relaxations of normalized graph-cut clustering (Xing & Jordan, 2003; Ng et al., 2001). By augmenting the k-means with randomized seeding technique, Arthur & Vassilvitskii (2007) obtained an algorithm $\Theta(\log k)$-competitive to the optimal. Unfortunately, all of these relaxations are restricted to Gaussian models of $P(\mathbf{X}|\mathbf{Y})$, and the optimization algorithms depend heavily on the linearity of the SDP objective.

The conditional clustering approach can be extended to hard *joint* clustering by explicitly including the class prior, thus optimizing the joint likelihood $P(\mathbf{X}, \mathbf{Y})$ with the most likely $\mathbf{Y}$. Again, efficient solution methods are not generally known, leaving local approaches as the only known option currently.

To smooth these objectives, the *soft joint* model optimizes the marginal likelihood, $P(\mathbf{X}) = \sum_Y P(\mathbf{Y})P(\mathbf{X}|\mathbf{Y})$ (Neal & Hinton, 1998; Banerjee et al., 2005), which has traditionally been tackled by the expectation-maximization (EM) algorithm (Dempster et al., 1977). The EM algorithm remains susceptible to local optima however. Intensive research has been devoted to understanding properties of the Gaussian mixture model in particular (Moitra & Valiant, 2010; Kalai et al., 2010; Dasgupta & Schulman, 2007; Chaudhuri et al., 2009). Although running time can be reduced to polynomial when the number of clusters or data dimensionality are constrained, it remains exponential in these quantities jointly. A few convex relaxations for soft joint clustering models have therefore been proposed. For example, Lashkari & Golland (2007) restrict cluster centers to data points, while Nowozin & Bakir (2008) impose sparsity inducing regularization over the class priors (while still embedding an intractable subproblem). Recent spectral techniques can provably recover an approximate estimate of Gaussian mixtures in polynomial time (Hsu & Kakade, 2013; Anandkumar et al., 2012). Unfortunately, this formulation remains restricted to spherical Gaussian models of $P(\mathbf{X}|\mathbf{Y})$.

Finally, *discriminative models* provide a distinct paradigm for clustering that can be more effective when the goal of learning is to predict labels from the observation $\mathbf{X}$, *e.g.* as in semi-supervised classification (Chapelle et al., 2006). In this approach, one maximizes the reverse conditional like-

lihood $P(\mathbf{Y}|\mathbf{X})$, with $\mathbf{Y}$ imputed by the most likely label. A straightforward optimization strategy can alternate between optimizing $\mathbf{Y}$ and the conditional model $P(\mathbf{Y}|\mathbf{X})$, but this quickly leads to local optima. Thus, here too, convex relaxation has been a popular approximation strategy, either in the case of a large margin loss (Xu & Schuurmans, 2005) or logistic loss (Joulin & Bach, 2012; Joulin et al., 2010; Bach & Harchaoui, 2007; Guo & Schuurmans, 2007). To date, such formulations have been entirely based on SDP relaxations with *unnormalized* equivalence matrices, whose elements indicate whether two examples belong to the same cluster. Such an approach is prone to discovering imbalanced clusters, since the model employs no natural mechanism that automatically avoids assigning all examples to a single cluster.

A word about the notation: bold faced uppercase variables, *e.g.* $\mathbf{X}, \mathbf{Y}$, are used to represent observed and latent variables respectively. Bold faced lowercase variables are used to denote vectors. Matrix variables are represented by uppercase alphabets, *e.g.* X, Y. For consistency, afterwards, we will use t to denote the number of data points, n to denote the dimension of each data point and d for the number of latent clusters.

2.2 Bregman Divergences

All of the loss models and probability models considered in this thesis will be based on Bregman divergences, which will therefore will play a key role in the clustering formulations I consider. A Bregman divergence defines a notion of dissimilarity between two points based on a strictly convex potential function. In particular, a Bregman divergence is defined as follows.

Definition 1. *Let $F : \mathcal{S} \mapsto \mathbb{R}, \mathcal{S} = \mathrm{dom}(F) \subseteq \mathbb{R}^n$, be a strictly convex function such that F is differentiable and $f = \nabla F$. The Bregman divergence $d_F(\mathbf{x}, \mathbf{y})$ is defined as*

$$d_F(\mathbf{x}, \mathbf{y}) := F(\mathbf{x}) - F(\mathbf{y}) - \langle \mathbf{x} - \mathbf{y}, f(\mathbf{y}) \rangle \,. \tag{2.1}$$

The strict convexity of F confers several important properties to d_F based on this definition. First, let $F^* : \mathbb{R}^n \mapsto \mathbb{R}$ be the Fenchel conjugate of F, *i.e.*

$$F^*(\mathbf{y}) = \sup_{\mathbf{x} \in \mathrm{dom}(F)} \mathbf{y}^\top \mathbf{x} - F(\mathbf{x}). \tag{2.2}$$

and let $f^* = \nabla F^*$. Since the strict convexity of F implies that f is invertible, we have that

$F(\mathbf{x})$	$f(\mathbf{x})$	$d_F(\mathbf{x},\mathbf{y})$	Divergence
$\mathbf{x}^2$	$\mathbf{x}$	$(\mathbf{x}-\mathbf{y})$	Squared loss
$\log(\frac{\mathbf{x}}{1-\mathbf{x}})$	$\mathbf{x}\log(\mathbf{x})+(1-\mathbf{x})\log(1-\mathbf{x})$	$\mathbf{x}\log(\frac{\mathbf{x}}{\mathbf{y}})+(1-\mathbf{x})\log(\frac{1-\mathbf{x}}{1-\mathbf{y}})$	Logistic loss
$-\frac{1}{\mathbf{x}}$	$-\log(\mathbf{x})$	$\frac{\mathbf{x}}{\mathbf{y}}-\log(\frac{\mathbf{x}}{\mathbf{y}})-1$	Itakura-Saito distance
$\mathbf{1}+\log(\mathbf{x})$	$\mathbf{x}^\top\log(\mathbf{x})$	$\mathbf{x}^\top\log(\frac{\mathbf{x}}{\mathbf{y}})$	KL-divergence

Table 2.1: Some Common Bregman Divergences

$F^*(\mathbf{y}) = \mathbf{y}^\top f^{-1}(\mathbf{y}) - F(f^{-1}(\mathbf{y}))$. Based on this observation, one can conclude that

$$\begin{aligned} f^*(\mathbf{y}) &= \nabla F^*(\mathbf{y}) \\ &= f^{-1}(\mathbf{y}) + J_{f^{-1}}(\mathbf{y})\mathbf{y} - J_{f^{-1}}(\mathbf{y})f(f^{-1}(\mathbf{y})) \\ &= f^{-1}(\mathbf{y}), \end{aligned} \tag{2.3}$$

where $J_{f^{-1}}$ is the Jacobian of f^{-1}. Note that the Definition 1 is the difference between a strictly convex function F at $\mathbf{x}$ and its first order Taylor approximation at another point $\mathbf{y}$. Based on this fact, several important properties follow.

1. Non-negativity. $d_F(\mathbf{x},\mathbf{y}) \geq 0$. This fact follows because a convex function necessarily dominates its first order Talyor approximation. Here equality is achieved if and only if $\mathbf{x} = \mathbf{y}$.

2. Convexity. d_F is always convex in the first argument, but not necessarily convex in the second argument.

3. Dual divergence. Given a strictly convex differentiable function F, and its invertible gradient function $f = \nabla F$ and conjugate function F^*, one can establish the following relationship between Bregman divergence and its dual divergence

$$\begin{aligned} d_F(\mathbf{x},\mathbf{y}) &= F(\mathbf{x}) - F(\mathbf{y}) - \langle \mathbf{x}-\mathbf{y}, f(\mathbf{y})\rangle \\ &= \left\langle f(\mathbf{x}), f^{-1}(f(\mathbf{x}))\right\rangle - F^*(f(\mathbf{x})) - \left\langle f(\mathbf{y}), f^{-1}(f(\mathbf{y}))\right\rangle + F^*(f(\mathbf{y})) \\ &\quad - \langle \mathbf{x}-\mathbf{y}, f(\mathbf{y})\rangle \\ &= F^*(f(\mathbf{y})) - F^*(f(\mathbf{x})) - \langle f(\mathbf{y}) - f(\mathbf{x}), \mathbf{y}\rangle \\ &= d_{F^*}(f(\mathbf{y}), f(\mathbf{x}))). \end{aligned} \tag{2.4}$$

Some examples of commonly used convex functions and their corresponding Bregman divergence are listed in Table 2.1. A more detailed discussion of other important properties of Bregman divergences are given in (Banerjee et al., 2005).

Since later on we will work closely with matrix notation, we will first introduce the following

notation:

$$D_F(A, B) = \sum_i d_F(A_{i:}, B_{i:}) \tag{2.5}$$

$$D_{F^*}(A, f(B)) = \sum_i d_{F^*}(A_{i:}, f(B_{i:})). \tag{2.6}$$

That is, we will write $D_F(A, B)$ to denote the sum of row-wise Bregman divergences. Also, with some abuse of notation, we will let $D_{F^*}(A, f(B))$ denote the sum of row-wise dual Bregman divergences. Throughout the thesis, whenever $f(\cdot)$ or $f^{-1}(\cdot)$ are applied to a matrix variable, we will assume these functions are applied row-wise.

2.2.1 Exponential Family Distributions

To explicitly model Bregman divergence clustering, we need to provide the definition of exponential family distributions, then we show the relationship between regular Bregman divergences and regular exponential family models.

Definition 2. *A probability density function or probability mass function $p(\mathbf{w}|\boldsymbol{\theta})$ for $\mathbf{w} \in \mathbb{R}^m$ and $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^n$ is in the exponential family, if it is of the form*

$$p(\mathbf{w}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} h(\mathbf{w}) \exp\left(\boldsymbol{\theta}^\top \phi(\mathbf{w})\right) \tag{2.7}$$

$$= h(\mathbf{w}) \exp\left(\boldsymbol{\theta}^\top \phi(\mathbf{w}) - A(\boldsymbol{\theta})\right), \tag{2.8}$$

for some functions ϕ and h, where

$$Z(\boldsymbol{\theta}) = \int h(\mathbf{w}) \exp\left(\boldsymbol{\theta}^\top \phi(\mathbf{w})\right) \tag{2.9}$$

$$A(\boldsymbol{\theta}) = \log Z(\boldsymbol{\theta}). \tag{2.10}$$

Here $\boldsymbol{\theta}$ are called **natural parameters**, *$\phi(\mathbf{w}) \in \mathbb{R}^n$ is called a vector of* **sufficient statistics**, *$Z(\boldsymbol{\theta})$ is called the partition function, $A(\boldsymbol{\theta})$ is called the* **log-partition function**, *and $h(\mathbf{w})$ is the scaling function, often 1.*

To simplify the subsequent statements, we follow (Banerjee et al., 2005) and define regular exponential family distributions through their minimal sufficient statistic $\mathbf{x} \in \mathbb{R}^n$.

Definition 3. *A regular exponential family is a multivariate parametric family of distributions where each probability density has the form*

$$p(\mathbf{x}|\boldsymbol{\theta}) = \exp(\boldsymbol{\theta}^\top \mathbf{x} - A(\boldsymbol{\theta})) p_0(\mathbf{x}), \forall \mathbf{x} \in \mathbb{R}^n. \tag{2.11}$$

Distribution	$p(\mathbf{x};\boldsymbol{\theta})$	$d_F(\mathbf{x},\mu)$
1-D Gaussian	$\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x-\theta)^2}{2\sigma^2}\right)$	$\frac{1}{2\sigma^2}(x-\mu)^2$
d-D Spherical Gaussian	$\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{\|\mathbf{x}-\boldsymbol{\theta}\|^2}{2\sigma^2}\right)$	$\frac{1}{2\sigma^2}\|\mathbf{x}-\boldsymbol{\mu}\|^2$
1-d Binomial	$\frac{N!}{(x)!(N-x)!}q^x(1-q)^{N-x}$	$xlog(\frac{x}{\mu})+(N-x)\log\left(\frac{N-x}{N-\mu}\right)$
d-D Multinomial	$\frac{N!}{\prod_{i=1}^d \mathbf{x}_i!}\prod_{i=1}^d q_i^{\mathbf{x}_i}$	$\mathbf{x}^\top\log\left(\frac{\mathbf{x}}{\boldsymbol{\mu}}\right)$

Table 2.2: Some popular exponential families and corresponding Bregman divergence.

The only difference between this definition and (2.8) is that we have embedded the feature function ϕ in the minimal sufficient statistic $\mathbf{x}$. Exponential family distributions and Bregman divergences are closely related. In fact, we have the following lemma from (Banerjee et al., 2005).

Lemma 1. *(Banerjee et al., 2005) If F^* is the log-partition function of a regular exponential family with natural parameter space $\Theta^* = \mathrm{int}(\mathrm{dom}(F^*))$, then*

1. *F^* is strictly convex on Θ^*, and its conjugate function F is also strictly convex on $\Theta = \mathrm{int}(\mathrm{dom}(F))$.*
2. *Both F and F^* are differentiable on Θ and Θ^* respectively. The gradient function $\nabla F = f$ is invertible and continuous, and $\nabla F = f = f^* = f^{-1}$.*

With this lemma, one can then establish the following relationship between regular exponential family distributions and Bregman divergences.

Theorem 2. *(Banerjee et al., 2005) Let*

$$p(\mathbf{x}|\boldsymbol{\theta}) = \exp\left(\boldsymbol{\theta}^\top\mathbf{x} - F^*(\boldsymbol{\theta})\right)p_0(\mathbf{x}) \tag{2.12}$$

be the probability density function of a regular exponential family distribution. Let F^* be the conjugate of F. Then $p(\mathbf{x}|\boldsymbol{\theta})$ can be uniquely expressed as

$$p(\mathbf{x}|\boldsymbol{\theta}) = \exp\left(-d_F(\mathbf{x}, f^{-1}(\boldsymbol{\theta}))Z_F(\mathbf{x})\right) \tag{2.13}$$

where $Z_F : \mathrm{dom}(F) \mapsto \mathbb{R}_+$ is a uniquely determined function.

More detailed discussion on bijections between regular exponential families and Bregman divergences can be found in (Banerjee et al., 2005).

This theorem shows that every regular exponential family corresponds to a unique and distinct Bregman divergence and every choice of Bregman divergence leads to a regular exponential family. Note that the proof of this bijection is not the contribution of this thesis. Some popular exponential family distributions and the corresponding Bregman divergences are given in Table 2.2.

2.3 Clustering Formulations based on Bregman divergences

Following (Banerjee et al., 2005), we formulate clustering as maximum likelihood estimation in an exponential family model with a latent variable $\mathbf{Y} \in \{1, \ldots, d\}$ (the class indicator). The observed variable $\mathbf{X}$ is in $\mathbb{R}^n$, from which an *iid* sample $X = (\mathbf{x}_1, \ldots, \mathbf{x}_t)'$ has been collected.

Generative models. In generative modeling we parameterize the joint distribution over $(\mathbf{X}, \mathbf{Y})$ as $\mathbf{Y} \to \mathbf{X}$:

$$p(\mathbf{Y} = j) = q_j, \tag{2.14}$$

$$p(\mathbf{X} = \mathbf{x} | \mathbf{Y} = j) = \exp\left(-d_F(\mathbf{x}, \boldsymbol{\mu}_j)\right) Z_j(\mathbf{x}). \tag{2.15}$$

Here $\Theta := \{q_j, \boldsymbol{\mu}_j\}_{j=1}^d$ are the parameters, where $\mathbf{q} \in \Delta_d$, the d dimensional simplex. We assume $P(\mathbf{X}|\mathbf{Y})$ is an exponential family model defined by the Bregman divergence d_F. Examples of commonly used Bregman divergences include Euclidean ($f(x) = x$), and sigmoid ($f(x) = \log \frac{x}{1-x}$).

Given data $X \in \mathbb{R}^{t \times n}$, the parameters Θ can be estimated via

$$\operatorname*{argmax}_{\Theta} \max_{Y} p(X, Y|\Theta) \tag{2.16}$$

$$\text{or} \quad \operatorname*{argmax}_{\Theta} \; p(X|\Theta) = \max_{\Theta} \sum_{Y} p(X, Y|\Theta), \tag{2.17}$$

depending on whether Y is to be maximized (hard clustering) or summed out (soft clustering). Here we are letting Y denote a $t \times d$ assignment matrix such that $Y_{ij} \in \{0, 1\}$ and $Y\mathbf{1} = \mathbf{1}$ (a vector of all 1's with proper dimension). If we additionally let $\Gamma = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_d)$ and $B = (\mathbf{b}_1, \ldots, \mathbf{b}_d)$, such that $\mathbf{b}_j = f(\boldsymbol{\mu}_j)$, then the conditional likelihood (2.15) can be rewritten over the entire data set as

$$p(X|Y) = \exp\left(-D_F(X, Y\Gamma)\right) Z(X) \tag{2.18}$$

$$= \exp\left(-D_{F^*}(YB, f(X))\right) Z(X), \tag{2.19}$$

where

$$D_F(X, Y\Gamma) := \sum_{i=1}^{t} d_F(X_{i:}, Y_{i:}\Gamma) \tag{2.20}$$

and

$$D_{F^*}(YB, f(X)) := \sum_{i=1}^{t} d_{F^*}(Y_{i:}B, f(X_{i:})) \tag{2.21}$$

are row-wise sums, such that $X_{i:}$ stands for the i-th row of X.

Discriminative models. As an alternative approach, discriminative clustering uses a graphical model $\mathbf{X} \rightarrow \mathbf{Y}$, and focuses on modeling the dependence of the labels Y given X:

$$p(Y|X; W, \mathbf{b}) = \exp(-D_{F^*}(Y, f(XW + \mathbf{1}\mathbf{b}'))) Z(X), \tag{2.22}$$

where $W \in \mathbb{R}^{n \times d}$ is the parameter to learn and $\mathbf{b} \in \mathbb{R}^d$ is the offset for all clusters. A soft clustering model cannot be applied in this case, since $\sum_Y p(X, Y) = p(X)$. Instead, hard optimization of Y leads to

$$\min_{W, \mathbf{b}, Y} D_F(XW + \mathbf{1}\mathbf{b}', f^{-1}(Y)). \tag{2.23}$$

All of these problems involve a mix of discrete and continuous variables, which raises considerable challenges. Our goal is to develop convex relaxations that can be solved efficiently while leading (after rounding) to higher quality solutions than those obtained by naive local optimization.

2.4 Generalized Conditional Gradient Method

All of the relaxations of clustering that will be investigated in this thesis reduce to optimization problems. To cope with these problems, I will develop scalable algorithms based on a simple but powerful optimization template, the generalized conditional gradient method. For the sake of completeness, I first provide some background on conditional gradient methods.

2.4.1 Conditional Gradient Method

Consider the optimization problem,

$$\min_{x \in \mathbf{Q}} f(x), \tag{2.26}$$

Algorithm 1 Conditional gradient method

1: Choose $x_0 \in \mathbf{Q}$ arbitrarily, set $k = 0$;
2: Solve

$$y_k \in \arg\min_{y \in \mathbf{Q}} \langle y, \nabla f(x_k) \rangle \,; \tag{2.24}$$

3: Perform a line-search by solving

$$\min_{\mu_k \in [0,1]} f\left((1-\mu_k)x_k + \mu_k y_k\right) \tag{2.25}$$

4: Assign $x_{k+1} \Leftarrow (1-\mu_k)x_k + \mu_k y_k$.

Algorithm 2 Generalized Conditional Gradient Method

1: Choose $x_0 \in \mathbf{H}$ such that $h(x_0) < \infty$ and set $k = 0$;
2: Solve

$$y_k \in \arg\min_{x \in \mathbf{H}} \langle y, \nabla f(x_k) \rangle + h(y); \tag{2.27}$$

3: Perform a line-search by solving

$$\min_{\mu_k \in [0,1]} f\left((1-\mu_k)x_k + \mu_k y_k\right) + h\left((1-\mu_k)x_k + \mu_k y_k\right); \tag{2.28}$$

4: Assign $x_{k+1} \Leftarrow (1-\mu_k)x_k + \mu_k y_k$.

where $\mathbf{Q}$ is a convex and bounded feasible region, and f is convex and smooth. The conditional gradient method is given in Algorithm 1. Note that, since finding y_k is actually a linear minimization problem, when $\mathbf{Q}$ is a polyhedra, (2.24) reduces to a linear program. Moreover, each iteration is well-defined because $\mathbf{Q}$ is bounded. Thus, each step of the algorithm involves a linear constrained minimization followed by a one-dimensional convex optimization, both of which are considered to be easy. However, the convergence rate is somehow slow (sublinear). It is also worth noting that the conditional gradient method is generally ineffective to apply in nonsmooth or stochastic settings.

2.4.2 Generalized Conditional Gradient Method

Although nonsmooth problems cause difficulty for the conditional gradient approach, a reasonable procedure can be achieved if the objective can be decomposed into a smooth and a nonsmooth part. A generalized conditional gradient method (K. Bredies & Maass, 2009) has been developed for this case.

Consider the following problem:

$$\min_{x \in \mathbf{Q}} f(x) + h(x) \tag{2.29}$$

where $\mathbf{Q}$ is convex and bounded such that f is smooth and convex while h is convex but not necessarily smooth.

The generalized conditional gradient algorithm is shown in Algorithm 2. With this algorithm, each step is now a smooth convex program, followed by a one dimensional nonsmooth convex optimization. Often, the smooth function h is a quadratic and Q is a polytope, making (2.27) a quadratic program. This algorithm can also be slow to converge to a global solution, since it has a sublinear rate (Zhang et al., 2012). A nice property of the generalized conditional gradient framework, however, is that many machine learning problems can be formulated in the form: a convex and differentiable loss function and a convex but not necessarily smooth regularization term.

2.4.3 Hybrid Approach: Local Search for Matrix-norm Regularization

The main optimization problems I consider in this thesis all involve optimization over matrix variables, where the regularization function h consists of a matrix norm, and challenging constraints such as positive semidefiniteness are also included. Therefore, the basic conditional gradient and generalized conditional gradient method have to be extended and improved to handle the matrix variable case.

Consider the convex optimization problem using matrix notation

$$\min_{X \in \mathbf{Q}} f(X) + h(X), \tag{2.30}$$

where X is a $n \times m$ matrix, $f(X) : \mathbb{R}^{n \times m} \to \mathbb{R}$ is a convex and smooth function, $h(X) : \mathbb{R}^{n \times m} \to \mathbb{R}$ is a convex but not necessarily smooth function, and $\mathbf{Q}$ is a convex feasible region. The clustering problems discussed in this thesis nicely fit into the generalized conditional gradient framework. The drawback of the generalized conditional gradient method is its sublinear rate. However, Zhang et al. (2012) observed that a fix-rank local optimization can be interlaced with the generalized conditional gradient procedure to significantly speed up the sparse learning model with matrix-norm regularization and semidefinite constraints. Specifically, when the matrix norm regularizer induces a low rank optimal matrix X, we can represent X with a low-rank factorization (say $X = UV^\top$) at each step.

Algorithm 3 Hybrid Generalized Conditional Gradient Method

1: One step conditional gradient on l.h.s. of (2.31);
2: Construct initialization for r.h.s. of (2.31);
3: Locally optimize over r.h.s. of (2.31);
4: Initialize l.h.s. of (2.31);

Then, the optimization problem can be reformulated as a hybrid algorithm that alternates between:

$$\min_{X} f(X) + h(X) \Leftrightarrow \min_{U,V} f(UV^\top) + h(UV^\top) \tag{2.31}$$

The hybrid generalized conditional gradient descent is outlined in Algorithm 3. For analysis and other applications, we refer to (Zhang et al., 2012) and (Laue, 2012) for more details.

This optimization strategy will allow scalable training methods to be developed for each of the convex relaxation schema developed later in this thesis.

Chapter 3

Conditional Generative Clustering Case 1: Jointly Convex Bregman Divergence

In this chapter, we first consider the case of *hard conditional clustering* for jointly convex Bregman divergences, where the class prior $\mathbf{q} \in \Delta_d$ has been fixed to some value in the d dimensional simplex beforehand.

Following the discussion in the background chapter, we formulate clustering as maximum likelihood estimation in an exponential family model with a latent variable $\mathbf{Y} \in \{1, \ldots, d\}$ (the class indicator). The observed variable $\mathbf{X}$ is in $\mathbb{R}^n$, from which an *iid* sample $X = (\mathbf{x}_1, \ldots, \mathbf{x}_t)'$ has been collected.

Recall that in generative modeling, we parameterize the joint distribution over $(\mathbf{X}, \mathbf{Y})$ as $\mathbf{Y} \to \mathbf{X}$, where

$$p(\mathbf{Y} = j) = q_j, \tag{3.1}$$

$$p(\mathbf{X} = \mathbf{x} | \mathbf{Y} = j) = \exp\left(-d_F(\mathbf{x}, \boldsymbol{\mu}_j)\right) Z_j(\mathbf{x}). \tag{3.2}$$

Because q_j is fixed to some value, here $\Theta := \{\boldsymbol{\mu}_j\}_{j=1}^d$ are the parameters to be optimized. We assume $P(\mathbf{X}|\mathbf{Y})$ is an exponential family model defined by the Bregman divergence D_F.

Since we focus on conditional generative hard clustering, given data X, the parameters Θ can be estimated via

$$\operatorname*{argmax}_{\Theta} \max_{Y} p(X, Y | \Theta). \tag{3.3}$$

Here we are letting Y denote a $t \times d$ assignment matrix such that $Y_{ij} \in \{0, 1\}$ and $Y\mathbf{1} = \mathbf{1}$ (a vector

of all 1's with proper dimension). If we additionally let $\Gamma = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_d)$ and $B = (\mathbf{b}_1, \ldots, \mathbf{b}_d)$, such that $\mathbf{b}_j = f(\boldsymbol{\mu}_j)$, then the conditional likelihood (3.2) can be rewritten over the entire data set as

$$p(X|Y) = \exp\left(-D_F(X, Y\Gamma)\right) Z(X) \tag{3.4}$$

$$= \exp\left(-D_{F^*}(YB, f(X))\right) Z(X), \tag{3.5}$$

where

$$D_F(X, Y\Gamma) := \sum_{i=1}^{t} d_F(X_{i:}, Y_{i:}\Gamma) \tag{3.6}$$

and

$$D_{F^*}(YB, f(X)) := \sum_{i=1}^{t} d_{F^*}(Y_{i:}B, f(X_{i:})) \tag{3.7}$$

are row-wise sums, such that $X_{i:}$ stands for the i-th row of X.

3.1 Formulation

First note that by using (3.4), the estimator (3.3) can be reduced to

$$\min_{Y,\Gamma} \; D_F(X, Y\Gamma). \tag{3.8}$$

Here Banerjee et al. (2005) showed that for any fixed assignment Y the optimal Γ is given by $\Gamma = (Y'Y)^{\dagger}Y'X$ ($\dagger$ denotes the pseudoinverse), for any Bregman divergence D_F. Plugging the solution back into the formulation, the problem becomes

$$\min_{Y} D_F(X, Y(Y'Y)^{\dagger}Y'X). \tag{3.9}$$

Now let us introduce the *normalized equivalence relation matrix*

$$M = Y(Y'Y)^{\dagger}Y' = Y\,\mathrm{diag}(Y'\mathbf{1})^{\dagger}Y', \tag{3.10}$$

and let $\mathcal{M}$ denote the set of possibilities. That is, $\mathcal{M} = \{M : \exists Y \in \{0,1\}^{t\times d}, Y\mathbf{1} = \mathbf{1}, M = Y(Y'Y)^{\dagger}Y'\}$. It then suffices to solve

$$\min_{M\in\mathcal{M}} D_F(X, MX). \tag{3.11}$$

This resulting problem is challenging for two reasons. First, the objective is not necessarily convex in M, since D_F is only guaranteed to be convex in its first argument. Another challenge lies in the non-convexity of the constraint set $\mathcal{M}$.

However, it is interesting that many widely used Bregman divergences are *jointly* convex in both arguments; *e.g.* Mahalonobis distance, KL divergence, Bernoulli entropy, Bose-Einstein entropy, Itakura-Saito distortion, and von Neumann divergence (Wang & Schuurmans, 2003; Tsuda et al., 2004). Therefore, we want to consider convex relaxations for the non-convex constraint set for conditional generative model clustering with jointly convex Bregman divergences first in this chapter, and then generalize the approach to arbitrary Bregman divergences in the next chapter.

Peng & Wei (2007) have shown that

$$\mathcal{M} = \left\{ M : M = M', M^2 = M, \mathrm{tr}(M) \leq d, M_{i:} \in \Delta_t \right\}.$$

First, note that since $M^2 = M$ is the only source of non-convexity, its convex hull can be used to construct a convex outer approximation $\mathcal{M}_1$ (*i.e.* convex containing sets) of the set $\mathcal{M}$:

$$\begin{aligned} \mathcal{M}_1 &:= \mathrm{conv}\left\{ M : M = M' = M^2 \right\} \cap \left\{ M \in \Delta_t^t : \mathrm{tr}(M) \leq d \right\} \\ &= \left\{ M : \mathbf{0} \preceq M \preceq I, \mathrm{tr}(M) \leq d, M_{i:} \in \Delta_t \right\}, \end{aligned} \tag{3.12}$$

where by $M \succeq \mathbf{0}$ we also encode $M = M'$. Note that $M \preceq I$ is implied by $\mathbf{0} \preceq M$ and $M_{i:} \in \Delta_t$ (*e.g.* Mirsky, 1955, Theorem 7.5.4).

Although this set $\mathcal{M}_1$ has been widely used, it is still not clear whether it is the tightest convex relaxation of $\mathcal{M}$; that is, whether $\mathcal{M}_1 = \mathrm{conv}\mathcal{M} = \mathcal{M}_C$? With some surprise, we show that this conjecture is not true in Appendix 8.1; that is, in general, $\mathcal{M}_1 - \mathcal{M}_C \neq \emptyset$. Despite the fact that $\mathcal{M}_1$ is a loose convex relaxation, its simplicity allows a simple and efficient optimization. Therefore, we will continue to use it below, as in the earlier work of (Peng & Wei, 2007). Conveniently, $\mathcal{M}_1$ can be relaxed further by keeping only the spectral constraints

$$\mathcal{M}_2 := \left\{ M : \mathbf{0} \preceq M \preceq I, \mathrm{tr}(M) \leq d, M\mathbf{1} = \mathbf{1} \right\}. \tag{3.13}$$

Therefore, based on the convex relaxation of constraint set $\mathcal{M}$ of normalized equivalence relation matrix, we managed to derive a convex relaxation for conditional generative model clustering with jointly convex Bregman divergences.

Algorithm 4 Alternating Direction Method of Multipliers

1: Repeat until convergence
2: $M_t \leftarrow \operatorname{argmin}_M \mathcal{L}(M, Z_{t-1}, \Lambda_{t-1})$;
3: $Z_t \leftarrow \operatorname{argmin}_Z \mathcal{L}(M_t, Z, \Lambda_{t-1})$;
4: $\Lambda_t \leftarrow \Lambda_{t-1} + \frac{1}{\rho}(Z_t - M_t)$;

3.2 Optimization

Here, since there is no matrix-norm regularization, instead of exploiting the specific optimization strategy we discussed in the previous chapter, we will temporarily need to adopt a slightly different optimization strategy. (Later chapters will revert to the same optimization strategy outlined originally.) Assuming D_F is convex in its second argument, one can easily minimize $D_F(X, MX)$ over $M \in \mathcal{M}_1$ by using the alternating direction method of multipliers (ADMM) (Boyd et al., 2010). In particular, we split the constraints into two groups: spectral and non-spectral, leading to the following augmented Lagrangian:

$$\begin{aligned}\mathcal{L}(M, Z, \Lambda) = D_F(X, MX) + \delta(M_{i:} \in \Delta_t) + \delta(Z \in \mathcal{M}_2) \\ - \langle \Lambda, M - Z \rangle + \frac{1}{2\rho} \|M - Z\|_F^2 ,\end{aligned}$$

where $\delta(\cdot) = 0$ if $\cdot$ is true; ∞ otherwise. The ADMM procedure then proceeds as follows: (i) optimize objective under non-spectral constraints; (ii) project to satisfy the spectral constraints; and (iii) update the multipliers; see Algorithm 4.

Note that since we constrain $M_{i:} \in \Delta_t$, the objective $D_F(X, MX)$ remains well defined in Step 1 of Algorithm 4. Furthermore, since the objective decomposes row-wise, each row of M can be optimized independently, which constitutes a key advantage of this scheme. Second, since Step 2 merely involves projection onto spectral constraints $\mathcal{M}_2$, a closed form solution exists based on eigen-decomposition, as established in the following lemma.

Lemma 3. *Let* $H = I - \frac{1}{t}\mathbf{1}\mathbf{1}'$. *Then*

$$\mathcal{M}_2 = \left\{ HMH + \tfrac{1}{t}\mathbf{1}\mathbf{1}' : M \in \mathcal{M}_3 \right\}, \tag{3.14}$$

$$\textit{where } \mathcal{M}_3 = \{ M : \mathbf{0} \preceq M \preceq I, \operatorname{tr}(M) \leq d - 1 \}. \tag{3.15}$$

Proof. Clearly the right-hand side of (3.14) is contained in $\mathcal{M}_2$. Conversely, for any $M_2 \in \mathcal{M}_2$, we

construct an $M \in \mathcal{M}_3$ as $M = M_2 - \frac{1}{t}\mathbf{1}\mathbf{1}'$. Note that $M_2\mathbf{1} = \mathbf{1}$ implies $\mathbf{1}/\sqrt{t}$ is an eigenvector of M_2 with eigenvalue 1. Therefore $M \succeq \mathbf{0}$. The rest is easy to verify. □

By Lemma 3, the problem of projecting any matrix A to $\mathcal{M}_2$ can be accomplished by solving

$$\min_{Z \in \mathcal{M}_2} \|Z - A\|^2 = \min_{S \in \mathcal{M}_3} \left\|HSH - (A - \tfrac{1}{t}\mathbf{1}\mathbf{1}')\right\|^2.$$

Let $B = A - \frac{1}{t}\mathbf{1}\mathbf{1}'$ and $V = B - HBH$. Then $HVH = \mathbf{0}$, hence the probem reduces to solving

$$\min_{S \in \mathcal{M}_3} \|HSH – HBH – V\|^2 = \min_{S \in \mathcal{M}_3} \|HSH – HBH\|^2 + \|V\|^2.$$

Now it suffices to solve $\min_{T \in \mathcal{M}_3} \|T - HBH\|^2$ *and* show the optimal T satisfies $HTH = T$. Suppose HBH has eigenvalues σ_i and eigenvectors ϕ_i. Then the optimal T must have eigenvalues μ_i and eigenvectors ϕ_i such that

$$\min_{\mu_i} \sum_i (\mu_i - \sigma_i)^2, \text{ s.t. } \mu_i \in [0, 1], \sum_i \mu_i \leq d - 1. \tag{3.16}$$

Since $\mathbf{1}$ is an eigenvector of HBH with eigenvalue 0, it is trivial that the corresponding μ_i in the optimal solution is also 0. Therefore, $T\mathbf{1} = \mathbf{0}$ and $HTH = T$. Finally the optimal Z is simply given by $T + \frac{1}{t}\mathbf{1}\mathbf{1}'$.

3.2.1 Rounding

Once an optimal solution is obtained for the relaxed problem, a feasible solution to the original problem can be obtained by heuristic rounding. Many rounding schemes can be applied with similar performance. Following the previous works (Guo & Schuurmans, 2007) and (Joulin & Bach, 2012), we apply spectral clustering (Shi & Malik, 2000) on M to obtain a rounded assignment matrix Y^*, *i.e.* using a k-means clustering on the eigenvectors associated with the k-largest eigenvalues. Then this Y^* is used to initialize an alternating hard EM procedure optimizing (3.8) to get a finer assignment matrix Y.

3.3 Experimental Evaluation

In this section, I evaluate the proposed convex relaxation for jointly convex Bregman divergence. In order to compare the proposed convex relaxation with the most related alternating hard EM algorithm, I use the common evaluation criteria: the objective value of (3.8) as well as the classification accuracy.

Dataset	t	n	d	Dataset	t	n	d
Yale	165	1024	15	Diabetes	768	8	2
ORL	400	1024	40	Heart	270	13	2
E-mail	1000	57	2	Breast	699	9	2
Balance	625	4	2				

Table 3.1: Properties of datasets used in experiment.

Data sets. I used seven labeled benchmark data sets for these experiments. Five are from the UCI repository (Frank & Asuncion, 2010): Balance, Breast Cancer, Diabetes, Heart, and Spam E-mail. The two others are multiclass face data sets: ORL[1] and Yale[2]. I downsampled Spam-Email to 1000 points preserving the class ratio. The properties of these data sets are summarized in Table 3.1, giving the values of t, n, and d. I shifted all features to be nonnegative so that all transfer functions can be applied. Finally the features were normalized to unit variance.

Transfer functions. Here, I tested two transfer functions: linear and sigmoid.

Parameter settings. The only parameter involved in the optimization algorithm is ρ for ADMM. With regard to efficiency and quality, it is set to 10^{-3}. The parameter selection is mainly based on time-efficiency. For more details on parameter tunning, see (Boyd et al., 2010).

Algorithms. The new proposed method (cvxCondJC) first minimizes $D_F(X, MX)$ over $M \in \mathcal{M}_1$. The optimal M is then rounded to a hard cluster assignment via spectral clustering (Shi & Malik, 2000). The result is further used to initialize a local re-optimization using the *original* objective $D_F(X, Y\Gamma)$. Since k-class spectral clustering involves a k-means algorithm, with random elements, this was repeated 10 times and the variance was reported.

I compared the new proposed algorithm with altCondJC (hard EM), which optimizes $D_F(X, Y\Gamma)$ by alternating, with Y reinitialized randomly up to the same time cost of our method with spectral clustering rounding and reoptimization.

Results. In Table 3.2, the first and third rows of each block gives the optimal value of $D_F(X, Y\Gamma)$ found by altCondJC, and by cvxCondJC (both after SC rounding and re-optimization). The second and fourth lines give the highest accuracy among all possible matchings between the clusters and ground truth labels. Across all data sets and transfer functions, cvxCondJC with SC rounding and reoptimization finds a lower objective value and higher accuracy than altCondJC. In addition,

[1] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
[2] http://cvc.yale.edu/projects/yalefaces/yalefaces.html

although the objective achieved after rounding might be higher than that of altCondJC, the accuracy is usually comparable. It is also worth noting that the accuracy of cvxCondJC with SC rounding is not necessarily improved by the reoptimization. Moreover, for data sets, ORL and Yale, which are image data sets, the accuracy of sigmoidal transfer function is higher than that of linear transfer function while for the rest datasets, the accuracy of linear transfer function would usually be higher (except Spam E-mail data set). It justifies the importance of different divergences for clustering. Overall, the final clustering found by cvxCondJC is superior to randomized local optimization over the evaluated data sets.

3.4 Conclusion

In this chapter, I have considered the conditional generative hard clustering with jointly convex Bregman divergences. In spite of its simplicity, this simple case actually provides insight into the formulation as well as a testbed to evaluate the quality of relaxation based on the normalized equivalence relation matrix. A key result is the analysis on the tightness of convex relaxation through normalized equivalence relation matrix. The resulting clustering formulation allows a distributed optimization procedure based on ADMM. On the basis of the experimental evaluation, it appears that the proposed new convex relaxation method outperforms local optimization both in terms of objective value and accuracy with the same time computational cost.

Unfortunately, Bregman divergences are not generally convex in their second argument, hence a more general relaxation strategy is required for convex relaxations of clustering with arbitrary Bregman divergence.

	cvxCondJC +SC rounding	cvxCondJC +SC+re-opt	altCondJC
	Spam E-mail		
lin_obj($\times 10^2$)	9.4± 0.1	**9.3**± 0.0	**9.3**± 0.0
lin_acc(%)	71.5±11.6	**76.3**±13.6	75.1±12.6
sigm_obj($\times 10^3$)	7.8± 0.1	**7.7**± 0.1	**7.7**± 0.1
sigm_acc(%)	75.1±12.0	**80.0**± 9.4	76.0± 7.2
	ORL		
lin_obj($\times 10^3$)	3.3± 0.1	**2.0**± 0.0	2.1± 0.0
lin_acc(%)	**57.0**± 3.5	55.4± 2.9	40.6± 2.3
sigm_obj($\times 10^2$)	3.8± 0.1	**3.5**± 0.1	3.7± 0.1
sigm_acc(%)	57.8± 3.6	**58.2**± 4.1	48.2± 3.0
	Yale		
lin_obj($\times 10^1$)	5.6± 0.1	**5.5**± 0.0	5.8± 0.1
lin_acc(%)	46.8± 1.7	**47.0**± 2.1	44.5± 4.2
sigm_obj($\times 10^2$)	9.6± 0.4	**9.2**± 0.1	9.6± 0.3
sigm_acc(%)	49.9± 2.1	**51.5**± 2.1	46.6± 4.1
	Balance		
lin_obj($\times 10^1$)	7.2± 0.0	**7.1**± 0.0	7.2± 0.0
lin_acc(%)	57.1± 6.9	**57.3**± 7.1	54.2± 4.6
sigm_obj($\times 10^2$)	5.0± 0.3	**3.9**± 0.0	4.0± 0.0
sigm_acc(%)	49.3± 5.1	**50.5**± 5.1	49.4± 4.3
	Breast Cancer		
lin_obj($\times 10^2$)	1.8± 0.2	**1.6**± 0.0	1.7± 0.0
lin_acc(%)	72.5±12.7	**84.7**± 8.8	78.7±10.4
sigm_obj($\times 10^2$)	**8.5**± 0.2	**8.5**± 0.1	**8.5**± 0.1
sigm_acc(%)	72.4±13.7	**72.5**±13.7	70.6±11.6
	Diabetes		
lin_obj($\times 10^2$)	**2.0**± 0.1	**2.0**± 0.0	**2.0**± 0.0
lin_acc(%)	57.1± 0.5	**58.5**± 0.0	**58.5**± 0.1
sigm_obj($\times 10^3$)	1.2± 0.1	**1.1**± 0.0	**1.1**± 0.0
sigm_acc(%)	**58.8**± 3.9	58.2± 0.1	58.0± 0.6
	Heart		
lin_obj($\times 10^2$)	**1.3**± 0.0	**1.3**± 0.0	**1.3**± 0.0
lin_acc(%)	**68.1**±10.0	65.6± 7.8	65.4± 5.0
sigm_obj($\times 10^2$)	7.5± 0.2	**7.2**± 0.2	**7.2**± 0.2
sigm_acc(%)	63.4± 5.9	**64.9**± 6.6	64.4± 7.8

Table 3.2: Experimental results for the conditional model with jointly convex Bregman divergences. Here "lin" and "sigm" refer to linear and sigmoid transfers respectively. Best results in **bold**.

Chapter 4

Conditional Generative Clustering Case 2: Arbitrary Bregman Divergence

Since a Bregman divergence is not necessarily convex in its second arugment, we need to extend the previous approach to consider the case of hard conditional clustering with arbitrary Bregman divergences.

In this chapter, we start as before and formulate clustering as maximum likelihood estimation in an exponential family model with a latent variable $\mathbf{Y} \in \{1, \ldots, d\}$ (the class indicator). Here, the observed variable $\mathbf{X}$ is in $\mathbb{R}^n$, from which an *iid* sample $X = (\mathbf{x}_1, \ldots, \mathbf{x}_t)'$ has been collected.

As before, we continue to consider a conditional generative model where the class prior $q \in \Delta_d$ has been fixed to some value in the d dimensional simplex beforehand. By assuming $P(X|Y)$ is an exponential family model defined by a Bregman divergence D_F, we can then reduce the estimation problem equivalently to

$$\min_{Y,B} \; D_{F^*}(YB, f(X)), \tag{4.1}$$

where $D_{F^*}(A, B) = \sum_i d_{F^*}(A_{i:}, B_{i:})$, d_{F^*} is the dual divergence of d_F, Y denotes a $t \times d$ assignment matrix such that $Y_{ij} \in \{0, 1\}$ and $Y\mathbf{1} = \mathbf{1}$.

4.1 Formulation

To cope with a general Bregman divergence, we need to adopt a significantly different strategy from from the last chaper. The key idea we exploit here is to introduce a value regularization that allows a useful form of representer theorem to be applied. In particular, we augment the negative log likelihood of $P(X|Y)$ in (4.1) with a regularizer on the basis B, weighted by the number of points

in the corresponding cluster. The resulting objective can be written as:

$$\min_{Y,B} D_{F^*}(YB, f(X)) + \frac{\alpha}{2} \|YB\|_F^2 . \tag{4.2}$$

The advantage of the formulation in Chapter 3 is that it does not require a regularizer, whereas the advantage of this formulation is that it allows more general loss functions.

Note that here B must be in the range of f. Now, by the representer theorem, there exists a matrix $A \in \mathbb{R}^{t \times n}$ such that the optimal B can be written $B = (Y'Y)^\dagger Y'A$ († denotes the pseudoinverse). Making this substitution in (4.2) yields

$$\min_{M \in \mathcal{M}, A} D_{F^*}(YB, f(X)) + \frac{\alpha}{2} \operatorname{tr}(A'MA), \tag{4.3}$$

where M is the *normalized equivalence relation matrix,* defined as before by

$$M = Y(Y'Y)^\dagger Y' = Y \operatorname{diag}(Y'\mathbf{1})^\dagger Y'. \tag{4.4}$$

In this chapter, instead of relaxing to the constraint set $\mathcal{M}_1$ in the previous chapter, we will work with this formulation by further relaxing the domain of M to the weaker convex set

$$\mathcal{M}_2 := \{M : \mathbf{0} \preceq M \preceq I, \operatorname{tr}(M) \leq d, M\mathbf{1} = \mathbf{1}\} . \tag{4.5}$$

The main motivation here is to develop a more efficient algorithm, because the existing polynomial time optimization algorithm in $\mathcal{M}_1$ is extremely slow in practice. By Lemma 3 in the last chaper, we can further relax the domain into $\mathcal{M}_3$

$$\mathcal{M}_3 := \{M : \mathbf{0} \preceq M \preceq I, \operatorname{tr}(M) \leq d - 1\}, \tag{4.6}$$

because $M_2 \in \mathcal{M}_2$ can be recovered from $M_3 \in \mathcal{M}_3$.

Due to the simplicity of the further relaxed constraint set $\mathcal{M}_3$, we will first develop convex relaxation strategy based on it and then extend it to $\mathcal{M}_2$.

First we establish the main optimization formulation that we will use in this chapter. Note that although (4.3) does not immediately exhibit joint convexity in M and A, a change of variable leads to a convex formulation. Denote $T = MA$. Then $\operatorname{Im}(T) \subseteq \operatorname{Im}(M)$ where $\operatorname{Im}(M)$ is the range of M. Also, denote $L(Z) := D_{F^*}(Z, f(X))$ for clarity.

Proposition 4. *The problem* (4.3) *is equivalent to*

$$\min_{M\in\mathcal{M}_3}\min_{T:\mathrm{Im}(T)\subseteq\mathrm{Im}(M)} L(T)+\frac{\alpha}{2}\operatorname{tr}(T'M^{\dagger}T) \tag{4.7}$$

$$=\min_{T} L(T)+\frac{\alpha}{2}\min_{M\in\mathcal{M}_3:\mathrm{Im}(T)\subseteq\mathrm{Im}(M)}\operatorname{tr}(T'M^{\dagger}T) \tag{4.8}$$

Proof. The proposition is easily established by observing that, any optimal (M, A) for (4.3) provides an optimal solution to (4.7) via $T = MA$. Conversely, given any optimal (M, T) for (4.7), This proposition allows one to solve a convex problem in T, provided that $\mathrm{Im}(T) \subseteq \mathrm{Im}(M)$ guarantees $T = MA$ for some A. □

4.2 Characterizing the regularizer in $\mathcal{M}_3$

Note that the optimization in (4.8) defines an implicit induced regularizer on T. Define

$$\Omega^2(T) := \min_{M\in\mathcal{M}_3:\mathrm{Im}(T)\subseteq\mathrm{Im}(M)}\operatorname{tr}(T'M^{\dagger}T), \tag{4.9}$$

which satisfies $\Omega(T) \geq 0$. The above proposition allows one to solve a convex problem in T, provided that $\Omega^2(T)$ is convex and easy to compute. Thus (M, A) is optimal for (4.3).

In order to better understand the computational complexity of $\Omega(T)$, in the following section, we will first try to characterize it $\Omega(T)$ computationally.

4.2.1 Efficient Computation of $\Omega(T)$

Let the singular values of T be $s_1 \geq \ldots \geq s_t$. Since $\Omega^2(T) = \min_{M\in\mathcal{M}_3}\operatorname{tr}(TT'M^{\dagger})$, by von Neumann's trace inequality (Mirsky, 1975) the optimal M must have eigenvectors equal to the left singular vectors of T. The minimal objective value is then $\sum_i s_i^2/\sigma_i$, where σ_i are the eigenvalues of M. It suffices to solve

$$f(\mathbf{s}) := \min_{\{\sigma_i\}}\sum_{i=1}^{t}\frac{s_i^2}{\sigma_i},\ \text{subject to}\ \sigma_i\in[0,1],\ \sum_{i=1}^{t}\sigma_i \leq d-1 \tag{4.10}$$

$$=\min_{\sigma_i\in[0,1]}\max_{\lambda\geq 0}\sum_{i=1}^{t}\frac{s_i^2}{\sigma_i}+\lambda\left(1-d+\sum_{i=1}^{t}\sigma_i\right) \tag{4.11}$$

$$=\max_{\lambda\geq 0}\left\{\lambda(1-d)+\min_{\sigma_i\in[0,1]}\sum_{i=1}^{t}\left(\frac{s_i^2}{\sigma_i}+\lambda\sigma_i\right)\right\}. \tag{4.12}$$

Fixing λ, the optimal σ_i is attained at $\sigma_i(\lambda) = \frac{s_i}{\sqrt{\lambda}}$ if $\lambda \geq s_i^2$, and 1 if $\lambda < s_i^2$. Note that $\sigma_i(\lambda)$ decreases monotonically for $\lambda \geq s_t^2$, hence we only need to find a λ that satisfies $\sum_{i=1}^{t}\sigma_i(\lambda) =$

Algorithm 5 Compute $f(\mathbf{s})$ with given d.

1: **for** $k = 0, 1, \ldots, d-2$ **do**
2: **if** $\sum_{i=k+1}^{t} s_i \geq (d-1-k)s_{k+1}$ **then** break
3: **end for**
4: **Return** $f(\mathbf{s}) = \sum_{i=1}^{k} s_i^2 + \frac{1}{d-1-k}\left(\sum_{i=k+1}^{t} s_i\right)^2$.

$d-1$, since the constraint $\sum_i \sigma_i \leq d-1$ must be equality at the optimum. This only requires a line search over λ, which can be conducted efficiently as follows. Suppose the optimal λ lies in $[s_k^2, s_{k+1}^2]$. Then $\sigma_i(\lambda) = 1$ for all $i \leq k$ and $\sigma_i(\lambda) = s_i/\sqrt{\lambda}$ for all $i > k$, so one can easily get the following condition from the optimality with respect to λ for (4.12)

$$k + \frac{1}{\sqrt{\lambda}} \sum_{i=k+1}^{t} s_i = d - 1. \tag{4.13}$$

Hence in order to find a $k \in \{1 \ldots t\}$ such that $s_{k+1}^2 \leq \lambda \leq s_k^2$ meeting (4.13), we just need to search $k = 1, \ldots, t$ and put $\sqrt{\lambda} = s_k$ and $\sqrt{\lambda} = s_{k+1}$ respectively back into (4.13) for a sign change, *i.e.*

$$\sqrt{\lambda} = s_k \Rightarrow k + \frac{\sum_{i=k+1}^{t} s_i}{s_k} \leq d-1,$$
$$\sqrt{\lambda} = s_{k+1} \Rightarrow k + \frac{\sum_{i=k+1}^{t} s_i}{s_{k+1}} \geq d-1.$$

Then the optimal λ can be computed as

$$\sqrt{\lambda} = \frac{1}{d-1-k} \sum_{i=k+1}^{t} s_i.$$

Now note there must be a k satisfying these two conditions. Since both $k + \frac{1}{s_k}\sum_{i=k+1}^{t} s_i$ and $k + \frac{1}{s_{k+1}}\sum_{i=k+1}^{t} s_i$ grow monotonically in k, the smallest k that satisfies the second condition must also satisfy the first condition. Hence the optimal solution is $\sigma_i = 1$ for all $i \leq k$, and $\sigma_i = (d-1-k)s_i / \sum_{i=k+1}^{t} s_i$ for $i > k$.

The algorithm for evaluating $f(\mathbf{s}) = \Omega^2(T)$, where $\mathbf{s}$ are the singular values of T, is given in Algorithm 5. The 'if' condition in step 2 must be met when $k = d-2$. The computational cost is dominated by a full SVD of T, and fortunately the proposed method needs to compute $\Omega(T)$ only once at the optimal T. Therefore, Ω can be computed in $O(t^3)$ time.

Interestingly, $\Omega(T)$ has other favorable properties to exploit.

Theorem 5. *$\Omega(T)$ defines a norm on T. Its dual norm is denoted as Ω_*.*

We prove this theorem in Appendix 8.2. Moreover, we also characterize its dual norm, which will be exploited later in the optimization strategy. Not surprisingly, the dual norm can also be computed efficiently with $O(t^2 d)$ time.

4.3 Extending the characterization of the regularizer from $\mathcal{M}_3$ to $\mathcal{M}_2$

Now we replace $\mathcal{M}_3$ in Proposition 4 by $\mathcal{M}_2$. In particular, we redo the characterization of $\Omega(T)$ when $\mathcal{M}_3$ is replaced by $\mathcal{M}_2$, and denote the new regularizer as $\Xi(T) \geq 0$, such that

$$\Xi^2(T) = \min_{M \in \mathcal{M}_2 : \mathrm{Im}(T) \subseteq \mathrm{Im}(M)} \mathrm{tr}(T' M^\dagger T). \tag{4.14}$$

If we can again show that $\Xi(T)$ is a norm such that both Ξ and the dual norm Ξ_* are efficiently computable, then the same optimization algorithm based on $\Omega(T)$ (given below in Section 4.4) can also be applied using Ξ without change. The remainder of this section proceeds in parallel to Section 4.2.

4.3.1 Efficient computation of $\Xi(T)$

First we apply Lemma 3 on page 18 us to convert the optimization in $\mathcal{M}_2$ into that in $\mathcal{M}_3$, making it easy to utilize the previous results.

Let

$$\mathrm{tr}(T' M^\dagger T) = \mathrm{tr}(Q M^\dagger), \tag{4.15}$$

where $Q = TT'$. To minimize it over $M \in \mathcal{M}_2$, by Lemma 3, it suffices to solve

$$\min_{M \in \mathcal{M}_3 : \mathrm{Im}(T) \subseteq \mathrm{Im}(HMH + \frac{1}{t}\mathbf{1}\mathbf{1}')} \mathrm{tr}\left(Q (HMH + \frac{1}{t}\mathbf{1}\mathbf{1}')^\dagger \right).$$

We first ignore the range constraint, and will show later that it will be automatically satisfied. Since $\mathbf{1}/\sqrt{t}$ is an eigen-vector of HMH with eigen-value 0, we have

$$\begin{aligned} (HMH + \frac{1}{t}\mathbf{1}\mathbf{1}')^\dagger &= (HMH)^\dagger + (\frac{1}{t}\mathbf{1}\mathbf{1}')^\dagger \\ &= (HMH)^\dagger + \frac{1}{t}\mathbf{1}\mathbf{1}'. \end{aligned} \tag{4.16}$$

By definition of H:

$$\begin{aligned} Q = IQI &= (H + \frac{1}{t}\mathbf{1}\mathbf{1}')Q(H + \frac{1}{t}\mathbf{1}\mathbf{1}') \\ &= HQH + \mathbf{1}\mathbf{q}'H + H\mathbf{q}\mathbf{1}' + s\mathbf{1}\mathbf{1}', \end{aligned} \tag{4.17}$$

where $\mathbf{q} := Q\mathbf{1}/t$ and $s := \mathbf{1}'\mathbf{q}/t = \mathbf{1}'Q\mathbf{1}/t^2$.

Next, we need to make use of the following lemma.

Lemma 6. *If $AB = \mathbf{0}$, then $A^\dagger B = \mathbf{0}$.*

Proof. Let $A = U\Sigma V'$ be the SVD of A. Then

$$\begin{aligned} AB = \mathbf{0} \;&\Rightarrow\; U\Sigma V'B = \mathbf{0} \;\Rightarrow\; \Sigma V'B = \mathbf{0} \\ \Rightarrow \Sigma^\dagger V'B = \mathbf{0} \;&\Rightarrow\; A^\dagger B = U\Sigma^\dagger V'B = \mathbf{0}. \end{aligned}$$

□

Similarly, if $BA = \mathbf{0}$ then $BA^\dagger = \mathbf{0}$.

Now, returning to (4.17), we note that since,

$$\begin{aligned} (HMH)(\mathbf{1}\mathbf{q}'H) &= \mathbf{0} \\ (HMH)(s\mathbf{1}\mathbf{1}') &= \mathbf{0} \\ (H\mathbf{q}\mathbf{1}')(HMH) &= \mathbf{0}, \end{aligned}$$

by Lemma 6, we have

$$\begin{aligned} (HMH)^\dagger(\mathbf{1}\mathbf{q}'H) &= \mathbf{0} \\ (HMH)^\dagger(s\mathbf{1}\mathbf{1}') &= \mathbf{0} \\ (H\mathbf{q}\mathbf{1}')(HMH)^\dagger &= \mathbf{0}. \end{aligned}$$

Therefore combining (4.16) and (4.17) we obtain

$$\begin{aligned} & \operatorname{tr}\left(Q(HMH + \frac{1}{t}\mathbf{1}\mathbf{1}')^\dagger\right) \\ = \;& \operatorname{tr}\left((HQH)(HMH)^\dagger\right) + \frac{1}{t}\mathbf{1}'Q\mathbf{1}. \end{aligned} \tag{4.18}$$

Clearly $HMH \in \mathcal{M}_3$ for any $M \in \mathcal{M}_3$. So if we find

$$M^* = \operatorname*{argmin}_{M \in \mathcal{M}_3} \operatorname{tr}\left((HQH)M^\dagger\right), \tag{4.19}$$

and show $M^* = HM^*H$, then M^* must be the minimizer of (4.18) over $M \in \mathcal{M}_3$. (4.19) is obviously in the same form as $\Omega^2(T) = \min_{M\in\mathcal{M}_3} \mathrm{tr}(TT'M^\dagger)$ and its optimal objective value is $\Omega^2(HT)$. By the discussion on how to compute Ω in Section 4.2, if HQH has eigenvectors ϕ_i with eigenvalue $\lambda_i > 0$, then

$$M^* = \sum_i \mu_i \phi_i \phi_i' \tag{4.20}$$

for some $\mu_i > 0$. Since $\mathbf{1}/\sqrt{t}$ is an eigenvector of HQH with eigenvalue 0, so $\phi_i'\mathbf{1} = 0$. Therefore $M^*\mathbf{1} = \mathbf{0}$ and $HM^*H = M^*$.

Finally we show $\mathrm{Im}(T) \subseteq \mathrm{Im}(HM^*H + \frac{1}{t}\mathbf{1}\mathbf{1}')$. By (4.20) and $HM^*H = M^*$, the nonzero eigenvectors[1] of $HM^*H + \frac{1}{t}\mathbf{1}\mathbf{1}'$ are $S := \{\mathbf{1}/\sqrt{t}\} \cup \{\phi_i\}_i$. So it suffices to show that S spans the left singular vectors of T, or equivalently the nonzero eigenvectors of Q. This means for any $\mathbf{u}$ that is orthogonal to $\mathbf{1}$ and ϕ_i, $Q\mathbf{u} = \mathbf{0}$. Since $Q \succeq \mathbf{0}$, we only need to show $\mathbf{u}'Q\mathbf{u} = 0$, which is obvious because by (4.17),

$$\begin{aligned}\mathbf{u}'Q\mathbf{u} &= \mathbf{u}'(HQH)\mathbf{u} + \mathbf{u}'\mathbf{1}\mathbf{q}'H\mathbf{u} + \mathbf{u}'H\mathbf{q}\mathbf{1}'\mathbf{u} + s\mathbf{u}'\mathbf{1}\mathbf{1}'\mathbf{u} \\ &= 0 + 0 + 0 + 0 + 0 = 0.\end{aligned}$$

To conclude,

$$\Xi^2(T) = \Omega^2(HT) + \frac{1}{t}\left\|T'\mathbf{1}\right\|^2, \tag{4.21}$$

$$M^* + \frac{1}{t}\mathbf{1}\mathbf{1}' = \underset{M\in\mathcal{M}_2:\mathrm{Im}(T)\subseteq\mathrm{Im}(M)}{\mathrm{argmin}} \mathrm{tr}(T'M^\dagger T). \tag{4.22}$$

Based on the discussion of the computional effciency above, we will have the following theorem on $\Xi(T)$.

Theorem 7. *$\Xi(T)$ defines a norm on T. Its dual norm is denoted as Ξ_*.*

Similar to the previous section, we will prove the theorem and characterize Ξ and its dual norm Ξ_* in Appendix 8.3.

[1] Eigenvectors whose corresponding eigenvalue is not zero

Algorithm 6 Conditional gradient for optimizing (4.8)

1: Initialize $T_0 = \mathbf{0}$. $s_0 = 0$.
2: **for** $k = 0, 1, \ldots$ **do**
3: Set $S_k \in \partial\Omega_*(\nabla L(T_k))$, *i.e.* find a minimizer of $\min_S \langle \nabla L(T_k), S\rangle + \frac{\alpha}{2}\Omega^2(S)$ up to scaling.
4: Line search:
$(a, b) := \operatorname{argmin}_{a\geq 0, b\geq 0} L(aT_k + bS_k) + \frac{\alpha}{2}(as_k + b)^2$.
5: Set $T_{k+1} = aT_k + bS_k$, $s_{k+1} = as_k + b$.
6: **end for**

4.4 Optimization

With these conclusions, we can optimize the primary objective (4.8) defined on page 25 using the generalized conditional gradient method, accelerated by local search (Laue, 2012; Zhang et al., 2012) as outlined in Chapter 2; see Algorithm 6.

At each iteration, the algorithm employs a linear approximation of L. The inner oracle searches for a steepest descent direction by computing a subgradient of the dual norm Ω_*, because when the maximum in (4.23) is achieved, every maximal value $\hat{\mathbf{x}}$ of (4.23) is a subgradient with respect to $\mathbf{z}$ of the conjugate function $f^*(\mathbf{z})$. To see why this is true, recall that generalized Legendre dual for non-differentiable functions is given by

$$f^*(\mathbf{z}) = \max_{\mathbf{x}} \mathbf{x}^\top \mathbf{z} - f(\mathbf{x}), \tag{4.23}$$

where $f^*(\mathbf{z})$ is the conjugate of the function $f(\mathbf{x})$, and the variable $\mathbf{z}$ is the dual variable of $\mathbf{x}$. Therefore, we must have

$$\partial f^*(\mathbf{z}) = \hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \mathbf{x}^\top \mathbf{z} - f(\mathbf{x}). \tag{4.24}$$

Algorithm 6 is guaranteed to find an ϵ accurate solution to (4.8) in $O(1/\epsilon)$ iterations; see e.g. (Zhang et al., 2012). The optimal M can then be recovered by evaluating Ω at the optimal T.[2]

4.4.1 Accelerated Hybrid Approach for Low-Rank Factorization

Due to Proposition 4, the norm regularizer in (4.8) induces a low rank optimal T. So if we explicitly represent T_k with a low-rank factorization (say $T_k = P_k Q_k'$ where P_k and Q_k have a small number of columns), then Ω_* (and its gradient) can be efficiently evaluated because a full SVD on T can be

[2]This solution is valid since (4.7) minimizes over M and T. If the problem were $\min_T \max_M$ instead, the optimal M could not be generally recovered by maximizing M for fixed optimal T.

performed efficiently by making use of such a low-rank factorization. For any vector $\mathbf{x}$, $T_k\mathbf{x}$ can be computed by $P_k(Q_k'\mathbf{x})$.

For (4.8), we can write $T = PQ'$ where P and Q have k columns (k is small). Then we can optimize over P and Q using any *local* solver and obtain any *local* solution. In practice, when k is large enough, there is a good chance that the solution is already very good.

Recall that at each iteration in Algorithm 6, S_k can be written as $\sum_{i=1}^{d} s_i\mathbf{u}_i\mathbf{v}_i'$. So after k iterations, T can be written as $\sum_{i=1}^{dk} s_i\mathbf{u}_i\mathbf{v}_i'$ (the set of $\mathbf{u}_i$ are not necessarily orthogonal, and neither are $\mathbf{v}_i$). If d and k are both small, this factorization will allow us to compute the full SVD of T efficiently. Therefore, based on low-rank factorization, the generalized conditional gradient method can be modified into Algorithm 7.

4.4.2 Extension to $\mathcal{M}_2$

By the discussion in Section 4.3, we can then extend the optimization procedure above from $\mathcal{M}_3$ to $\mathcal{M}_2$ as Algorithm 8.

4.4.3 Rounding

Once an optimal solution $M^* \in \mathcal{M}_2$ is obtained for the relaxed problem (4.3), a feasible solution to the original problem (4.1) can be obtained by heuristic rounding. Many rounding schemes can be applied with similar performances. Following previous works (Guo & Schuurmans, 2007) and (Joulin & Bach, 2012), we apply spectral clustering (Shi & Malik, 2000) on M to obtain a rounded assignment matrix Y^*, *i.e.* using a k-means clustering on the eigenvectors associated with the k-largest eigenvalues. Then this Y^* is used to initialize an alternating hard EM procedure optimizing (4.1) to get a finer assignment matrix Y.

4.5 Experimental Evaluation

In this section, I evaluate the proposed convex relaxation for conditional generative model with arbitrary Bregman divergence with the same data sets and transfer functions as the previous chapter. In order to compare the proposed convex relaxation with the most related alternating hard EM algorithm, I use the same evaluation criteria as before, the objective value of (4.1) as well as the

Algorithm 7 Accelerated Hybrid Optimization Procedure for Low-Rank Factorization in $\mathcal{M}_3$

Initialize $T_0 = \mathbf{0}, P_0 = Q_0 = []$(Matlab empty matrix), $r_0 = 0$
for $k = 1, 2, \ldots$ **do**
  Compute the gradient of L at T_{k-1}: $G = \nabla L\left(T_{k-1}, X\right)$.
  Generate weak hypothesis

$$S_k = \underset{T:\Omega(T)\leq 1}{\operatorname{argmin}} \operatorname{tr}(G'T) = -\underset{T:\Omega(T)\leq 1}{\operatorname{argmax}} \operatorname{tr}(G'T). \tag{4.25}$$

  By (4.24) and the discussion in Section 4.2, S_k can be written as $\sum_{i=1}^{d-1} s_i \mathbf{u}_i \mathbf{v}_i'$.
  Check termination criteria
  if $\operatorname{tr}(G'S_k) + \alpha r_{k-1} > -\epsilon$ **then**
    break
  end if
  Partially corrective update

$$\{\eta_1^*, \eta_2^*\} := \underset{\eta_1,\eta_2\geq 0}{\operatorname{argmin}} L\left(\eta_1 T_{k-1} + \eta_2 S_k, X\right) + \frac{\alpha}{2}\left(\eta_1 r_{k-1} + \eta_2\right)^2. \tag{4.26}$$

  Locally solve

$$\min_{P,Q} L(PQ') + \frac{\alpha}{2}\Omega^2(PQ') \tag{4.27}$$

  by initializing

$$P = (\sqrt{\eta_1^*}P_{k-1}, \sqrt{\eta_2^* s_1}\mathbf{u}_1, \ldots, \sqrt{\eta_2^* s_{d-1}}\mathbf{u}_{d-1})$$
$$Q = (\sqrt{\eta_1^*}Q_{k-1}, \sqrt{\eta_2^* s_1}\mathbf{v}_1, \ldots, \sqrt{\eta_2^* s_{d-1}}\mathbf{v}_{d-1})$$

  Denote the locally optimal solution as (P_k, Q_k).
  Set the solution at iteration k: $T_k = P_k Q_k'$. Restore r_k by solving

$$r_k = \min_{\eta_i, S_i:\eta_i\geq 0, \Omega(S_i)\leq 1, \sum_i \eta_i S_i = T_k} \sum_i \eta_i = \Omega(T_k). \tag{4.28}$$

  This is actually the gauge function of the unit ball of Ω evaluated at T_k. So trivially $r_k = \Omega(T_k)$ (which matches our intuition).
end for
Return T_k

classification accuracy.

Parameter settings. To closely approximate the original objective without creating numerical difficulty, I choose the regularization parameter α to be reasonably small $\alpha \in \{10^{-5}, 10^{-9}\}$ and report the experimental results for the choices that obtain highest accuracy. However, the results are not sensitive to these values.

Algorithms. The new proposed method (cvxCond) first optimize (4.3) over $M \in \mathcal{M}_2$. Then

Algorithm 8 Accelerated Hybrid Optimization Procedure for Low-Rank Factorization in $\mathcal{M}_2$

Initialize $T_0 = \mathbf{0}, P_0 = Q_0 = []$(Matlab empty matrix), $r_0 = 0$
for $k = 1, 2, \ldots$ **do**
  Compute the gradient of L at T_{k-1}: $G = \nabla L\left(T_{k-1}, X\right)$.
  Generate weak hypothesis

$$S_k = \underset{T:\Xi(T)\leq 1}{\operatorname{argmin}} \operatorname{tr}(G'T) = -\underset{T:\Xi(T)\leq 1}{\operatorname{argmax}} \operatorname{tr}(G'T). \tag{4.29}$$

  By (4.24) and the discussion in Section 4.3, we have compact way to represent

$$\begin{aligned} S_k &= -(aHUS + b\mathbf{1}\mathbf{1}'U\Sigma)V' \\ &= -(\tilde{a}HU + b\mathbf{1}\mathbf{1}'U)\Sigma V' \qquad (4.30) \\ &= -\tilde{U}\Sigma V', \qquad (4.31) \end{aligned}$$

  where $a = \sqrt{1 - \frac{(\tau^*)^2}{t}}$, $\tilde{a} = \frac{1}{\|\operatorname{diag}(\Sigma)\|}\sqrt{1 - \frac{(\tau^*)^2}{t}}$, $b = \frac{\tau^*}{t\|G'\mathbf{e}\|}$, the top $d-1$ SVD of $G = U\Sigma V'$ and $S = \Sigma/\|\operatorname{diag}(\Sigma)\|$. Thus, $S_k = \sum_{i=1}^{d-1}\sigma_i\tilde{\mathbf{u}}_i\mathbf{v}_i'$.
  Check termination criteria
  if $\operatorname{tr}(G'S_k) + \alpha r_{k-1} > -\epsilon$ **then**
    break
  end if
  Partially corrective update

$$\{\eta_1^*, \eta_2^*\} := \underset{\eta_1,\eta_2\geq 0}{\operatorname{argmin}} L\left(\eta_1 T_{k-1} + \eta_2 S_k, X\right) + \frac{\alpha}{2}\left(\eta_1 r_{k-1} + \eta_2\right)^2. \tag{4.32}$$

  Locally solve

$$\min_{P,Q} L(PQ') + \frac{\alpha}{2}\Xi^2(PQ') \tag{4.33}$$

  by initializing

$$\begin{aligned} P &= (\sqrt{\eta_1^*}P_{k-1}, \sqrt{\eta_2^* s_1}\tilde{\mathbf{u}}_1, \ldots, \sqrt{\eta_2^* s_{d-1}}\tilde{\mathbf{u}}_{d-1}) \\ Q &= (\sqrt{\eta_1^*}Q_{k-1}, \sqrt{\eta_2^* s_1}\mathbf{v}_1, \ldots, \sqrt{\eta_2^* s_{d-1}}\mathbf{v}_{d-1}) \end{aligned}$$

  Denote the locally optimal solution as (P_k, Q_k).
  Set the solution at iteration k: $T_k = P_k Q_k'$. Restore r_k by solving

$$r_k = \min_{\eta_i, S_i:\eta_i\geq 0, \Xi(S_i)\leq 1, \sum_i \eta_i S_i = T_k} \sum_i \eta_i = \Xi(T_k). \tag{4.34}$$

end for
Return T_k

similar to Section 3.3, the optimal M is rounded by spectral clustering (10 repeats). Here subsequent re-optimization (based on local optimization) is performed on the objective $D_{F^*}(YB, f(X))$. The competing algorithm, altCond, optimizes this objective by alternating with random initializations

of Y up to the same time-cost of cvxCondwith SC rounding and reoptimization.

Results. In Table 4.1, the first and third rows of each block gives the optimal value of $D_{F^*}(YB, f(X))$ found by altCond, and by cvxCond (both after SC rounding and re-optimization). The second and fourth rows give the highest accuracy among all possible matchings between the clusters and ground truth labels. Here it can be observed that for almost all data sets and transfer functions, cvxCond with SC rounding and reoptimization yields lower optimal objective value and higher accuracy than altCond, except two outliers, Diabetes and Heart with sigmoidal transfer function. Moreover, the objective values also exhibit lower standard deviation than altCond, which suggests that the value regularization scheme helps stabilize the reoptimization. It is also worth noting that the accuracy of cvxCondJC with SC rounding is not necessarily improved by the reoptimization. For data sets, ORL and Yale, which are image data sets, the accuracy of sigmoidal transfer function is higher than that of linear transfer function while for the rest datasets, the accuracy of linear transfer function would usually be higher. It justifies the importance of different divergences for clustering. Note that the accuracy of cvxCond with rounding is already comparable with that of altCond on most data sets.

Since the same transfer functions as Chapter 3 are used here, the accuracy of cvxCondJC with SC rounding and reoptimization is really close to those of cvxCond with SC rounding and reoptimization. It indicates that the extra regularization we employ for convex relaxation would not decimate the performance.

4.6 Conclusion

In this chapter, we have developed a more general convex relaxation strategy for conditional generative hard clustering with arbitrary Bregman divergence.

The key idea we apply is to introduce the normalized equivalence relation matrix by applying a value regularization. An important technique that will be widely exploited afterwards, using an induced matrix norm to promote low-rank, is developed to enable direct application of generalized conditional gradient method, accelerated by local search. Based on the experimental evaluation, the proposed new method performs better in terms of both objective value and accuracy than the corresponding local alternate algorithm with the same time cost.

So far, the clustering probability models we have considered are solely conditional generative

models based only on $P(X|Y)$. However, in practice, discriminative models with the reverse conditional $P(Y|X)$ have been proved to be very accurate data-driven tools for learning the input variables and the latent labels. Therefore, we will consider extending our techniques to convex relaxation for discriminative models in the following chapter.

	cvxCond	cvxCond +SC rounding	cvxCond +SC rounding & re-opt	altCond
	Spam E-mail			
lin_obj($\times 10^2$)	**9.3**± 0.1	**9.3**± 0.0	**9.3**± 0.0
lin_acc(%)	75.0± 9.0	**79.8**±10.2	73.9±13.3
sigm_obj($\times 10^3$)	8.0± 0.2	**7.7**± 0.1	**7.7**± 0.1
sigm_acc(%)	64.8±12.5	**78.7**± 7.8	75.3± 5.5
	ORL			
lin_obj($\times 10^3$)	2.7± 0.1	**2.0**± 0.0	2.1± 0.0
lin_acc(%)	**62.6**± 3.0	59.4± 2.4	40.1± 2.3
sigm_obj($\times 10^2$)	4.0± 0.1	**3.4**± 0.0	3.7± 0.1
sigm_acc(%)	**60.1**± 6.1	60.0± 4.9	48.6± 2.7
	Yale			
lin_obj($\times 10^1$)	6.1± 0.2	**5.7**± 0.1	5.8± 0.1
lin_acc(%)	43.3± 3.2	**45.2**± 3.2	44.4± 4.0
sigm_obj($\times 10^2$)	10.3± 0.2	**9.3**± 0.1	9.5± 0.2
sigm_acc(%)	46.6± 2.6	**51.1**± 2.7	46.2± 3.0
	Balance			
lin_obj($\times 10^1$)	8.0± 0.4	**7.1**± 0.0	**7.1**± 0.0
lin_acc(%)	57.1± 6.9	**57.3**± 7.1	55.5± 5.1
sigm_obj($\times 10^2$)	4.0± 0.0	**3.9**± 0.0	4.0± 0.1
sigm_acc(%)	**54.1**± 8.3	53.0± 6.0	50.9± 5.2
	Breast Cancer			
lin_obj($\times 10^2$)	1.7± 0.1	**1.6**± 0.0	1.7± 0.0
lin_acc(%)	75.4±13.3	**85.8**± 6.6	78.7±10.9
sigm_obj($\times 10^2$)	8.8± 0.2	**8.5**± 0.1	8.6± 0.2
sigm_acc(%)	66.8± 8.4	**72.3**±12.5	70.3±11.0
	Diabetes			
lin_obj($\times 10^2$)	**2.0**± 0.0	**2.0**± 0.0	**2.0**± 0.0
lin_acc(%)	58.1± 0.6	**58.3**± 0.0	58.2± 0.1
sigm_obj($\times 10^3$)	1.2± 0.1	1.1± 0.0	**1.0**± 0.0
sigm_acc(%)	54.7± 3.0	**58.2**± 0.2	58.1± 0.5
	Heart			
lin_obj($\times 10^2$)	**1.3**± 0.0	**1.3**± 0.0	**1.3**± 0.0
lin_acc(%)	**69.4**± 9.3	67.0± 5.5	66.1± 5.2
sigm_obj($\times 10^2$)	7.2± 0.1	**7.1**± 0.1	7.3± 0.2
sigm_acc(%)	**66.9**±10.7	64.9± 8.2	65.8± 6.3

Table 4.1: Experimental results for the conditional model with arbitrary Bregman divergences. Best results shown in **bold**.

Chapter 5

Discriminative Clustering

Although generative models can often reveal useful latent structure in data, many problems such as semi-supervised learning and multiple instance learning are more concerned with accurate label prediction. In such settings, discriminative models $\mathbf{X} \to \mathbf{Y}$ can often be more effective (Joulin & Bach, 2012; Bach & Harchaoui, 2007; Xu & Schuurmans, 2005). Therefore, in this chapter, we will consider convex relaxation for this setting.

As before, we formulate clustering as maximum likelihood estimation in an exponential family model with a latent variable $\mathbf{Y} \in \{1, \ldots, d\}$ (the class indicator). The observed variable $\mathbf{X}$ is in $\mathbb{R}^n$, from which an *iid* sample $X = (\mathbf{x}_1, \ldots, \mathbf{x}_t)'$ has been collected. Unlike generative model, discriminative clustering uses a graphical model $\mathbf{X} \to \mathbf{Y}$, and focuses on modeling the dependence of the labels Y given X:

$$p(Y|X; W, \mathbf{b}) = \exp(-D_{F^*}(Y, f(XW + \mathbf{1}\mathbf{b}'))) Z(X),$$

where W is the parameter to learn and $\mathbf{b} \in \mathbb{R}^d$ is the offset for all clusters. A soft clustering model cannot be applied in this case, since $\sum_Y p(X, Y) = p(X)$. Instead, hard partition optimization of Y leads to

$$\min_{W, \mathbf{b}, Y} D_F(XW + \mathbf{1}\mathbf{b}', f^{-1}(Y)). \tag{5.1}$$

Unlike generative models, for discriminative clustering, we only consider a special case where potential function $F(\mathbf{x}) = \log \sum_i \exp(x_i)$, *i.e.* where the transfer function $f = \nabla F$ is sigmoidal (Joulin & Bach, 2012). The reason for this is because sigmoidal transfer function satisfies the multinomial conditional model for the class indicator.

5.1 Formulation

Before attempting a convex relaxation for the discriminative model (5.1), it is important to recognize that a plain optimization over $(W, \mathbf{b}, Y)$ using the sigmoidal transfer will lead to vacuous solutions, where all examples are assigned to a single cluster j and $b_j = \infty$. A common solution is to add a regularizer on Y to enforce a more balanced cluster distribution. A natural choice of regularizer on Y is the entropy of cluster sizes, *i.e.* $-h(Y'\mathbf{1})$ where $h(\mathbf{x}) = \sum_i x_i \log x_i$. Note that this situation is opposite to generative clustering, where one must upper bound d, since otherwise the joint likelihood would be trivially maximized by assigning each data point to its own cluster.

In the following , we derive a convex relaxation for discriminative clustering based on the formulation

$$\min_{W,\mathbf{b},Y} \frac{1}{t} D_F(XW + \mathbf{1}\mathbf{b}', f^{-1}(Y)) + h(Y'\mathbf{1}). \tag{5.2}$$

The key idea is to do so using a *normalized equivalence relation matrix* in this setting.

By adding value regularization $\|WY'\|^2$ to (5.2), one obtains

$$\min_{W,\mathbf{b},Y} \frac{1}{t} D_F(XW + \mathbf{1}\mathbf{b}', f^{-1}(Y)) + \frac{\gamma}{2} \left\|WY'\right\|^2 + h(Y'\mathbf{1}). \tag{5.3}$$

Then expanding the Bregman divergence according to its definition, we can reformulate the above problem equivalently as

$$\begin{aligned} &\min_{W,\mathbf{b},Y} \frac{1}{t} F(XW + \mathbf{1}\mathbf{b}') - \frac{1}{t} \operatorname{tr}((XW + \mathbf{1}\mathbf{b}')Y') \\ &\qquad - \frac{1}{t} F(Y) + \frac{\gamma}{2} \left\|WY'\right\|^2 + h(Y'\mathbf{1}) \qquad (5.4) \\ = &\min_{W,\mathbf{b},Y} \max_{\Lambda:\Lambda_{i:} \in \Delta} -\frac{1}{t} F^*(\Lambda) + \frac{1}{t} \operatorname{tr}(\Lambda'(XW + \mathbf{1}\mathbf{b}')) \\ &\qquad - \frac{1}{t} \operatorname{tr}((XW + \mathbf{1}\mathbf{b}')Y') - \frac{1}{t} F(Y) + \frac{\gamma}{2} \left\|WY'\right\|^2 + h(Y'\mathbf{1}). \qquad (5.5) \end{aligned}$$

Here, based on Fenchel's identity $F(\mathbf{x}) = \max_{\mathbf{z} \in \operatorname{dom} F^*} \mathbf{x}'\mathbf{z} - F^*(\mathbf{z})$ where dom denotes the effective domain of a convex function, the second step follows from replacing $F(XW + \mathbf{1}\mathbf{b}')$ with its Fenchel conjugate.

Then, by applying a change of variable, $\Lambda = \Omega Y$, and converting the constraints on Λ to $\Omega_{i:} \in \Delta$

(Guo & Schuurmans, 2007), one can get the following equivalent problem

$$\min_{W,\mathbf{b},Y}\max_{\Omega:\Omega_i\in\Delta} -\frac{1}{t}F^*(\Omega Y)+\frac{1}{t}\operatorname{tr}(Y'\Omega'(XW+\mathbf{1}\mathbf{b}')) \\ -\frac{1}{t}F(Y)-\frac{1}{t}\operatorname{tr}((XW+\mathbf{1}\mathbf{b}')Y')+\frac{\gamma}{2}\left\|WY'\right\|^2+h(Y'\mathbf{1}). \tag{5.6}$$

Moreover, the outer minimization with respect to W and $\mathbf{b}$ can be achieved by setting

$$W=\tfrac{1}{t}X'(I-\Omega)Y(Y'Y)^\dagger, \text{ and } \Omega'\mathbf{1}=\mathbf{1}. \tag{5.7}$$

Note that $-\frac{1}{t}F^*(\Omega Y)+h(Y'\mathbf{1})\leq -\frac{1}{t}F^*(\Omega)+c_0$ where c_0 is some constant (Joulin & Bach, 2012, Eq 3). Using (5.7) and the fact that $F(Y)$ is a constant, one can upper bound (5.6) by

$$\min_{M\in\mathcal{M}}\max_{\Omega:\Omega_i\in\Delta,\Omega'\mathbf{1}=\mathbf{1}} -\frac{1}{t}F^*(\Omega)-\frac{1}{2\gamma t^2}\left\|X'(I-\Omega)M\right\|^2. \tag{5.8}$$

Importantly, this formulation is expressed completely in terms of the normalized equivalence relation matrix M, which constitutes a significant advantage over (Joulin & Bach, 2012; Guo & Schuurmans, 2007). Rather than resort to the proximal gradient method to solve for Ω given M (Joulin & Bach, 2012), which is slow in practice, we can harness the power of second order solvers like L-BFGS by dualizing the problem back to the primal form, which leads to an unconstrained problem. This reformulation also sheds light on the nature of the relaxation (5.8).

5.2 Optimization

Fixing $M\in\mathcal{M}$, we add a Lagrange multiplier $\boldsymbol{\tau}\in\mathbb{R}^t$ to enforce $\Omega'\mathbf{1}=\mathbf{1}$. By introducing the change of variable $\Psi=I-\Omega$, the optimization over Ω becomes equivalent to

$$\min_{\Psi\leq I:\Psi\mathbf{1}=\mathbf{0}} \frac{1}{t}F^*(I-\Psi)+\frac{1}{2\gamma t^2}\left\|X'\Psi M\right\|^2+\frac{1}{t}\boldsymbol{\tau}'\Psi\mathbf{1}. \tag{5.9}$$

The tool we use for dualization is provided by the following lemma.

Lemma 8. (Borwein & Lewis, 2000, Theorem 3.3.5) *Let J and G be convex functions, and A a linear transform. Suppose $A\operatorname{dom}J$ has nonempty intersection with $\{\mathbf{x}\in\operatorname{dom}G^* : G^* \text{ is continuous at } \mathbf{x}\}$. Then*

$$\min_{\mathbf{x}} J(\mathbf{x})+G(A\mathbf{x})=\max_{\mathbf{y}} -J^*(-A'\mathbf{y})-G^*(\mathbf{y}). \tag{5.10}$$

To apply Lemma 8 to (5.9), choose the linear transform A to be $\Psi \mapsto \frac{1}{t}X'\Psi M$, $G(\Psi) = \frac{1}{2\gamma}\operatorname{tr}(\Psi M^\dagger \Psi')$,[1] and $J(\Psi) = \frac{1}{t}F^*(I-\Psi) + \frac{1}{t}\boldsymbol{\tau}'\Psi\mathbf{1}$ over $\Psi\mathbf{1} = \mathbf{0}$ and $\Psi \leq I$ (elementwise). Then the problem (5.9) becomes equivalent to

$$\min_{M,\boldsymbol{\tau},\Upsilon\in\mathbb{R}^{t\times n}} \frac{1}{t}\sum_i [F(\tfrac{1}{t}X_{i:}\Upsilon'M + \boldsymbol{\tau}') - (\tfrac{1}{t}X_{i:}\Upsilon'M_{:i} + \tau_i)] + \frac{\gamma}{2}\operatorname{tr}(\Upsilon'M\Upsilon). \tag{5.11}$$

Note that $F(\mathbf{x}) = \log\sum_i \exp(x_i)$ can be interpreted as a soft max, hence the result is related to the typical max-margin style model. The loss of each example i is the soft max of $X_{i:}\Upsilon'M + \boldsymbol{\tau}'$ (a row vector) minus $X_{i:}\Upsilon'M_{:i} + \tau_i$. Here τ_i is an offset associated with each training example (cf. b_j for each cluster).

The most straightforward method for optimizing (5.11) is to treat it as a convex function of M, whose gradient and objective value can be evaluated by minimizing out Υ and $\boldsymbol{\tau}$. Since both Υ and $\boldsymbol{\tau}$ are unconstrained, this can be easily accomplished by quasi-Newton methods like L-BFGS. Interestingly, thanks to the structure of the problem, we can optimize (5.11) even more efficiently by applying the same change of variable as in Section 4.4. Letting $V = M\Upsilon \in \mathbb{R}^{t\times n}$ and constraining M to $\mathcal{M}_3 = \{M : \mathbf{0} \preceq M \preceq I, \operatorname{tr}(M) \leq d-1\}$, the problem (5.11) becomes

$$\min_{V,\boldsymbol{\tau}} \frac{\gamma}{2}\Omega^2(V) + \frac{1}{t}\sum_i [F(\tfrac{1}{t}X_{i:}V' + \boldsymbol{\tau}') - (\tfrac{1}{t}X_{i:}V'_{i:} + \tau_i)]. \tag{5.12}$$

Denote

$$L(V) = \frac{1}{t}\sum_i [F(\tfrac{1}{t}X_{i:}V' + \boldsymbol{\tau}') - (\tfrac{1}{t}X_{i:}V'_{i:} + \tau_i)]. \tag{5.13}$$

The objective (5.12) again absorbs the spectral constraints on M into the norm Ω, and can be readily solved by generalized conditional gradient in Algorithm 9. The extension to $M \in \mathcal{M}_2 = \{M : \mathbf{0} \preceq M \preceq I, \operatorname{tr}(M) \leq d, M\mathbf{1} = \mathbf{1}\}$ is also immediate.

5.2.1 Rounding

Once the optimal solution is obtained for the relaxed problem, a feasible solution to the original problem can be achieved by heuristic rounding. Many rounding schemes can be applied with similar performance. Following previous works (Guo & Schuurmans, 2007) and (Joulin & Bach, 2012), we apply spectral clustering (Shi & Malik, 2000) on M to obtain a rounded assignment matrix Y^*, *i.e.* using a k-means clustering on the eigenvectors associated with the k-largest eigenvalues.

[1] Since $M^2 = M$ for $M \in \mathcal{M}$, (5.9) can also be recovered by setting $G(\Psi) = \frac{1}{2\gamma}\operatorname{tr}(\Psi\Psi')$. However, to reformulate the problem into (5.12), which is the key to efficient optimization, it is crucial to include $M^\dagger$ in G.

Algorithm 9 Conditional gradient for optimizing (5.12)

1: Initialize $V_0 = \mathbf{0}$. $s_0 = 0$.
2: **for** $k = 0, 1, \ldots$ **do**
3: Set $S_k \in \partial\Omega_*(\nabla L(V_k))$, *i.e.* find a minimizer of $\min_S \langle \nabla L(V_k), S\rangle + \frac{\alpha}{2}\Omega^2(S)$ up to scaling.
4: Line search:
$(a, b) := \operatorname{argmin}_{a\geq 0, b\geq 0} L(aV_k + bS_k) + \frac{\alpha}{2}(as_k + b)^2$.
5: Set $V_{k+1} = aV_k + bS_k$, $s_{k+1} = as_k + b$.
6: **end for**

5.3 Experimental Evaluation

In this section, I evaluate the proposed convex relaxation for discriminative model through normalized equivalence relation matrix with the same data sets. In order to compare the proposed convex relaxation with two previous convex relaxations (Guo & Schuurmans, 2007) and (Joulin & Bach, 2012) , I use the same evaluation criteria as before, the objective value of (5.2) as well as the classification accuracy.

Parameter settings. To closely approximate the original objective without creating numerical difficulty, we choose the regularization parameter γ to be reasonably small $\gamma \in \{10^{-6}, 10^{-9}\}$ and report the experimental results for the choices that obtain highest accuracy. Again, the results are not sensitive to these values.

Algorithms. The new proposed method (cvxDisc) optimizes (5.11) over $M \in \mathcal{M}_2$ by solving (5.12). I also test on the algorithms of Joulin & Bach (2012) and Guo & Schuurmans (2007), which we refer to as JB and GS. The result of all the three methods are rounded by spectral clustering, then used to initialize a local re-optimization over (5.2). Since the discriminative model is logistic, we use the sigmoid transfer in D_F only.

Results. According to Table 5.1, cvxDisc with SC rounding only already achieves higher or comparable accuracy to both JB and GS in most cases except Diabetes and Heart. Further improvements can be obtained by reoptimization for all data set but Balance. For JB, it only performs really well for Diabetes while GS performs the best for Breast Cancer. Regarding the runtime for solving the respective convex relaxations, cvxDisc is at least 10 times faster than both JB and GS. This confirms the computational advantage of our primal reformulation (5.11), compared to other implementations of convex relaxation. Therefore, in terms of accuracy and runtime, the proposed cvxDisc is superior to the other two.

	cvxDisc	JB	GS
	Spam E-mail		
run time ($\times 10^4$s)	**0.005**	0.651	2.148
obj w/ SC rounding ($\times 10^3$)	**8.0**± 0.2	8.7± 0.0	8.2± 0.2
obj w/ SC + re-opt ($\times 10^3$)	**7.6**± 0.0	7.9± 0.2	**7.6**± 0.0
acc w/ SC rounding (%)	**69.9**±14.3	60.7± 0.1	62.8± 9.2
acc w/ SC + re-opt (%)	**83.5**± 7.8	61.3± 9.2	81.4± 5.6
	ORL		
run time ($\times 10^4$s)	**0.080**	0.695	6.372
obj w/ SC rounding ($\times 10^2$)	4.1± 0.1	7.1± 0.0	**3.6**± 0.0
obj w/ SC + re-opt ($\times 10^3$)	**3.5**± 0.0	3.8± 0.1	3.6± 0.0
acc w/ SC rounding (%)	**59.4**± 2.7	20.0± 1.1	54.6± 2.1
acc w/ SC + re-opt (%)	**59.5**± 2.8	45.2± 2.5	54.6± 2.4
	Yale		
run time ($\times 10^3$s)	**0.050**	0.648	6.745
obj w/ SC rounding ($\times 10^3$)	**8.6**± 0.2	13.2± 0.0	10.2± 0.3
obj w/ SC + re-opt ($\times 10^3$)	**7.6**± 0.1	8.3± 0.1	7.8± 0.3
acc w/ SC rounding (%)	**44.3**± 2.5	16.2± 0.6	33.8± 3.6
acc w/ SC + re-opt (%)	**46.1**± 2.9	34.1± 2.6	42.4± 2.7
	Balance		
run time ($\times 10^4$s)	**0.004**	0.155	0.078
obj w/ SC rounding ($\times 10^2$)	5.1± 0.0	6.1± 0.0	**4.9**± 0.1
obj w/ SC + re-opt ($\times 10^2$)	**3.9**± 0.0	4.5± 0.0	4.1± 0.2
acc w/ SC rounding (%)	**62.0**± 2.3	47.0± 1.8	46.5± 6.3
acc w/ SC + re-opt (%)	58.7± 0.0	**62.3**± 1.8	52.2± 5.2
	Breast Cancer		
run time ($\times 10^4$s)	**0.006**	0.479	1.758
obj w/ SC rounding ($\times 10^2$)	**8.5**± 0.0	10.0± 0.0	9.1± 0.2
obj w/ SC + re-opt ($\times 10^2$)	**8.4**± 0.0	8.7± 0.3	**8.4**± 0.1
acc w/ SC rounding (%)	**79.8**±15.7	60.4± 3.6	72.3±10.3
acc w/ SC + re-opt (%)	80.7±12.5	60.0± 4.2	**84.4**± 8.8
	Diabetes		
run time ($\times 10^4$s)	**0.012**	1.722	2.731
obj w/ SC rounding ($\times 10^3$)	**1.2**± 0.1	1.4± 0.0	1.3± 0.1
obj w/ SC + re-opt ($\times 10^3$)	**1.1**± 0.0	**1.1**± 0.0	**1.1**± 0.0
acc w/ SC rounding (%)	53.5± 3.1	**64.8**± 0.0	56.6± 4.2
acc w/ SC + re-opt (%)	58.3± 0.2	**58.6**± 0.0	58.3± 0.2
	Heart		
run time ($\times 10^4$s)	**0.001**	0.212	6.848
obj w/ SC rounding ($\times 10^2$)	**7.6**± 0.4	8.6± 0.0	7.7± 0.4
obj w/ SC + re-opt ($\times 10^3$)	**7.3**± 0.3	7.9± 0.0	**7.3**± 0.2
acc w/ SC rounding (%)	61.7± 5.8	55.2± 0.0	**64.4**± 9.5
acc w/ SC + re-opt (%)	**66.0**± 5.7	51.1± 0.0	65.2± 8.4

Table 5.1: Experimental results for the discriminative models.

Compared with conditional generative clustering (cvxCond and cvxCondJC) discussed in the previous two chapters, cvxDisc can indeed achieve higher accuracy for most data sets except Yale. It actually confirms that discriminative models usually would be more efficient in learning more accurate label prediction.

5.4 Conclusion

In this chapter, we have considered the case of discriminative clustering. By applying a value regularization, we derived a convex relaxation for discriminative clustering that uses the normalized equivalence relation matrix. A significant advantage over previous convex relaxations of discriminative clustering with unnormalized equivalence relation is that this new formulation promotes more balanced clusters and avoids vacuous results.

Moreover, for the optimization process, we can harness the power of second order solvers in the unconstrained primal form leading to more efficient algorithm. The experimental evaluation not only shows that the proposed method is significantly faster than other recent approaches for discriminative clustering, but it also enjoys comparable and even superior performance both in terms of objective value and accuracy.

Chapter 6

Joint Generative Clustering

In all generative models considered so far, we have ignored the cluster prior $\mathbf{q}$. This quantity is often useful in practice for inference at the cluster level, and can often be learned well by joint generative models. Therefore, in this chapter, we will extend our convex relaxation techniques to this setting.

As before, we formulate clustering as maximum likelihood estimation in an exponential family model with a latent variable $\mathbf{Y} \in \{1, \dots, d\}$ (the class indicator). The observed variable $\mathbf{X}$ is in $\mathbb{R}^n$, from which an *iid* sample $X = (\mathbf{x}_1, \dots, \mathbf{x}_t)'$ has been collected.

We turn to generative modeling, and parameterize the joint distribution over $(\mathbf{X}, \mathbf{Y})$ as $\mathbf{Y} \to \mathbf{X}$:

$$p(\mathbf{Y} = j) = q_j, \tag{6.1}$$

$$p(\mathbf{X} = \mathbf{x} | \mathbf{Y} = j) = \exp\left(-D_F(\mathbf{x}, \boldsymbol{\mu}_j)\right) Z_j(\mathbf{x}). \tag{6.2}$$

Here $\Theta := \{q_j, \boldsymbol{\mu}_j\}_{j=1}^d$ are the parameters, where $\mathbf{q} \in \Delta_d$, the d dimensional simplex. Again, We assume $P(\mathbf{X}|\mathbf{Y})$ is an exponential family model defined by the Bregman divergence D_F. Then, given data X, the conditional likelihood (6.2) can be rewritten as

$$p(X|Y) = \exp\left(-D_F(X, Y\Gamma)\right) Z(X) \tag{6.3}$$

$$= \exp\left(-D_{F^*}(YB, f(X))\right) Z(X), \tag{6.4}$$

where Y denote a $t \times d$ assignment matrix such that $Y_{ij} \in \{0, 1\}$ and $Y\mathbf{1} = \mathbf{1}$ (a vector of all 1's with proper dimension), $\Gamma = (\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_d)$ and $B = (\mathbf{b}_1, \dots, \mathbf{b}_d)$, such that $\mathbf{b}_j = f(\boldsymbol{\mu}_j)$.

6.1 Formulation

Different from conditional generative models, here we assume a multinomial distribution over cluster prior parameterized by $\mathbf{w} \in \mathbb{R}^d$:

$$p(\mathbf{Y} = j) = \exp(w_j - g(\mathbf{w})) \tag{6.5}$$

where

$$g(\mathbf{w}) = \log \sum_i \exp(x_i). \tag{6.6}$$

Then by (6.1) and (6.4), the negative log joint likelihood is:

$$-\mathbf{1}'Y\mathbf{w} + tg(\mathbf{w}) + L(YB) + \text{const}, \tag{6.7}$$

where

$$L(YB) = D_{F^*}(YB, f(X)). \tag{6.8}$$

Same as before, we can add regularizers on $\mathbf{w}$ and B, as well as an entropic regularizer $h(Y'\mathbf{1})$ to encourage cluster diversity, yielding:

$$\begin{aligned} \min_{\mathbf{w},B,Y} -\frac{1}{t}\mathbf{1}'Y\mathbf{w} + g(\mathbf{w}) + \frac{\beta}{2}\left\|Y\mathbf{w}\right\|^2 + h(Y'\mathbf{1}) \\ + \frac{1}{t}L(YB) + \frac{\alpha}{2}\left\|YB\right\|_F^2. \end{aligned} \tag{6.9}$$

This formulation can be convexified in terms of M by using the same techniques as Chapter 4 and 5, respectively. In particular, consider the prior $p(Y)$ as a discriminative model $Z \to Y$, where Z can only take a constant scalar value 1. Then, the first line of (6.9) is equivalent to

$$\frac{1}{t}g(Z\mathbf{1}\mathbf{w}') - \frac{1}{t}\operatorname{tr}(Z\mathbf{1}\mathbf{w}'Y') + \frac{\beta}{2}\left\|\mathbf{1}\mathbf{w}'Y'\right\|^2 + h(Y'\mathbf{1}). \tag{6.10}$$

By treating Z as the X in Chapter 5, it is easy to show that the first line of (6.9) can be upper bounded by (ignoring the offset $\boldsymbol{\tau}$):

$$\min_{\mathbf{s}\in\mathbb{R}^t} \frac{\beta}{2}\operatorname{tr}(\mathbf{s}'M\mathbf{s}) - \frac{1}{t}\mathbf{1}'M\mathbf{s} + g\left(\frac{1}{t}M\mathbf{s}\right). \tag{6.11}$$

Finally by applying the same technique that converted (4.2) to (4.3) in conditional generative model, one can reformulate (6.9) into:

$$\min_{A,M,\mathbf{s}} \frac{\beta}{2}\operatorname{tr}(\mathbf{s}'M\mathbf{s}) - \frac{1}{t}\mathbf{1}'M\mathbf{s} + g(\tfrac{1}{t}M\mathbf{s}) + \frac{1}{t}L(MA) + \frac{\alpha}{2}\operatorname{tr}(A'MA). \tag{6.12}$$

Therefore, by applying convex relaxation techniques developed previous in this thesis, we are able to derive a convex relaxation for joint generative model clustering with arbitrary Bregman divergences through normalized equivalence relations. With explicit control over the number of clusters, the proposed new method can take advantage of an efficient optimization procedure based on recent development of matrix learning.

6.2 Optimization

To optimize this formulation, let $\mathbf{u} = M\mathbf{s} \in \mathbb{R}^t$ and $T = MA \in \mathbb{R}^{t\times n}$. Then with $M \in \mathcal{M}_3 = \{M : \mathbf{0} \preceq M \preceq I, \operatorname{tr}(M) \leq d-1\}$, (6.12) becomes

$$\min_{\mathbf{u},T} g\Big(\frac{\mathbf{u}}{t}\Big) - \frac{1}{t}\mathbf{1}'\mathbf{u} + \frac{1}{t}L(T) + \min_{M\in\mathcal{M}_3}\frac{\beta}{2}\mathbf{u}'M^\dagger\mathbf{u} + \frac{\alpha}{2}\operatorname{tr}(T'M^\dagger T).$$

Denote $S := [\sqrt{\beta}\mathbf{u}, \sqrt{\alpha}T]$. Then $\operatorname{Im}(T) \subseteq \operatorname{Im}(M)$ and $\mathbf{u} \subseteq \operatorname{Im}(M)$ are equivalent to $\operatorname{Im}(S) \subseteq \operatorname{Im}(M)$. So

$$\alpha\operatorname{tr}(T'M^\dagger T) + \beta\operatorname{tr}(\mathbf{u}'M^\dagger\mathbf{u}) \tag{6.13}$$

$$= \operatorname{tr}((\alpha TT' + \beta\mathbf{u}\mathbf{u}')M^\dagger) \tag{6.14}$$

$$= \operatorname{tr}(SS'M^\dagger) = \operatorname{tr}(S'M^\dagger S). \tag{6.15}$$

By the same argument as in Proposition 4, the above problem can be reformulated as

$$\min_{\mathbf{u},T} \Gamma([\sqrt{\beta}\mathbf{u}, \sqrt{\alpha}T]) + \frac{1}{2}\Omega^2([\sqrt{\beta}\mathbf{u}, \sqrt{\alpha}T]) \tag{6.16}$$

$$= \min_S \Gamma(S) + \frac{1}{2}\Omega^2(S), \tag{6.17}$$

where

$$\Gamma(S) = \Gamma([\sqrt{\beta}\mathbf{u}, \sqrt{\alpha}T]) = g\left(\frac{\mathbf{u}}{t}\right) - \frac{1}{t}\mathbf{1}'\mathbf{u} + \frac{1}{t}L(T). \tag{6.18}$$

which can be solved by the methods developed previously and the algorithm is outlined in Algorithm 10. The extension to $\mathcal{M}_2 := \{M : \mathbf{0} \preceq M \preceq I, \operatorname{tr}(M) \leq d, M\mathbf{1} = \mathbf{1}\}$ is straightforward.

Algorithm 10 Conditional gradient for optimizing (6.16)

1: Initialize $S_0 = \mathbf{0}$. $g_0 = 0$.
2: **for** $k = 0, 1, \ldots$ **do**
3: Set $G_k \in \partial\Omega_*(\nabla L(S_k))$, *i.e.* find a minimizer of $\min_G \langle \nabla L(S_k), G\rangle + \frac{\alpha}{2}\Omega^2(G)$ up to scaling.
4: Line search:
$(a, b) := \operatorname{argmin}_{a\geq 0, b\geq 0} L(aS_k + bG_k) + \frac{\alpha}{2}(ag_k + b)^2$.
5: Set $S_{k+1} = aS_k + bG_k$, $g_{k+1} = ag_k + b$.
6: **end for**

6.2.1 Rounding

Once an optimal solution is obtained for the relaxed problem, a feasible solution to the original problem can be obtained by heuristic rounding. Many rounding schemes can be applied with similar performance. Following previous works (Guo & Schuurmans, 2007) and (Joulin & Bach, 2012), we apply spectral clustering (Shi & Malik, 2000) on M to obtain a rounded assignment matrix Y^*, *i.e.*using a k-means clustering on the eigenvectors associated with the k-largest eigenvalues. Then this Y^* is used to initialize an alternating procedure optimizing (6.7) to get a finer assignment matrix Y.

6.3 Experimental Evaluation

I evaluate the proposed convex relaxation for joint generative model through *normalized* equivalence relations with the same datasets and transfer functions as conditional generative models. Besides the common criteria classification accuracy, we also define the soft accuracy to compare different methods.

Parameters settings. In order to closely approximate the original objective without creating numerical difficulty, I choose the regularization parameter α and β to be reasonably small $\alpha \in \{10^{-5}, 10^{-9}\}$, $\beta \in \{10^{-5}, 10^{-9}\}$ and report the experimental results with highest accuracy. As before, the results are not sensitive to these values.

Algorithms. The proposed new method, cvxJoint, optimizes (6.12) over $M \in \mathcal{M}_2$ by solving (6.16). As before, I round the optimal M by spectral clustering to get the assignment matrix Y, and use it to initialize local reoptimization of the joint likelihood (6.7).

I compare the results to those of three soft generative models. The standard soft EM (Banerjee

et al., 2005, Algorithm 3) is randomly reinitialized 20 times. The other two algorithms are LG (Lashkari & Golland, 2007), and NB[1] (Nowozin & Bakir, 2008). Since they do not directly control the number of clusters, I tune their parameters so that the resulting number of cluster is d, or a little higher than d which could be truncated based on the cluster prior.

Results. Since joint models also learn a cluster prior, accuracy can take two forms. The hard accuracy is computed by $\operatorname{argmax}_y p(y|\mathbf{x}_i) = \operatorname{argmax}_y p(y)p(\mathbf{x}_i|y)$ in the case of soft EM, LG, and NB. Our model outputs a hard accuracy by locally reoptimizing the joint likelihood. For all methods, we define the soft accuracy based on the posterior distribution: $\max_\pi \mathbb{E}_{Y\sim p(Y|X)}[\text{Accuracy}(Y, \pi(Y^*))]$, where Y^* is the ground truth label and π is a matching between the cluster and label.

As shown in Table 6.1, for linear transfer function, cvxJoint with rounding and reoptimization achieves superior performance to the competing algorithms, both in terms of hard *and* soft accuracy, except Balance data set. For sigmoidal transfer function, cvxJoint with rounding and reoptimization achieves better performance for most data sets except Yale and Balance. The reason for LG to achieve better performance on Balance for two transfer functions is probably exemplar center might be more efficient for this data set. Moreover, the local reoptimization does not necessarily help achieve improvement in both accuracy and soft accuracy.

Compared with conditional models discussed in previous chapters, cvxJoint with rounding and reoptmization achieves higher accuracy for ORL, Yale, and Diabetes. For the rest of the data sets, the performance of joint model is really close to that of conditional models.

6.4 Conclusion

In this chapter, we consider the joint generative model which takes the cluster prior $\mathbf{q}$ into consideration. By assuming a multinomial distribution over cluster prior and applying the same value and entropic regularizer, we extend our convex relaxation technique to this setting. Compared with closely related joint clustering approaches, our model achieves empirical superior or comparable performance in term of hard and soft accuracy.

[1] http://www.nowozin.net/sebastian/infex. Since their approach relies heavily on the Gaussian model, I put NA in the corresponding cells in Table 6.1.

	linear		sigmoid	
	acc(%)	soft acc(%)	acc(%)	soft acc(%)
	Spam E-mail			
cvxJoint1	55.7±1.9	55.9±1.4	62.6±9.0	67.7±11.0
cvxJoint2	**60.5**±0.0	**60.5**±0.0	**81.5**±16.4	**79.2**±15.1
softEM	**60.5**±0.0	54.5±2.6	58.2±7.4	52.9±2.0
LG	60.0	0.1	40.6	1.8
NB	**60.5**	51.4	NA	NA
	ORL			
cvxJoint1	**61.0**±1.3	52.6±1.5	**63.0**±2.3	58.6±1.8
cvxJoint2	55.9±1.4	**52.8**±1.2	58.7±2.7	**58.7**±2.7
softEM	39.6±2.1	37.0±2.0	44.9±3.1	44.7±3.1
LG	40.0	1.9	36.0	0.5
NB	12.0	5.3	NA	NA
	Yale			
cvxJoint1	**47.9**±3.8	**45.9**±3.1	61.9±8.3	55.9±1.4
cvxJoint2	45.8±3.4	45.1±3.1	60.5±0.0	**60.5**±0.0
softEM	39.6±2.1	37.0±2.0	60.5±0.0	**60.5**±0.0
LG	35.2	4.8	**66.9**	0.1
NB	20.6	10.4	NA	NA
	Balance			
cvxJoint1	50.5±2.3	36.3±0.7	51.6±2.7	39.5±1.2
cvxJoint2	46.1±0.0	46.1±0.0	46.1±0.0	**46.1**±0.0
softEM	46.1±0.0	38.1±2.8	46.1±0.0	39.6±0.0
LG	**57.4**	0.2	**59.0**	0.2
NB	54.2	**54.7**	NA	NA
	Breast Cancer			
cvxJoint1	**71.0**±11.9	56.9±4.7	**70.9**±13.0	63.9±8.1
cvxJoint2	65.5±0.0	**65.5**±0.0	65.5±0.0	**65.5**±0.0
softEM	65.5±0.0	57.7±4.5	65.5±0.0	55.5±5.4
LG	61.8	0.1	65.5	0.1
NB	69.8	50.3	NA	NA
	Diabetes			
cvxJoint1	56.0±2.6	53.6±2.5	57.5±5.5	57.6±5.6
cvxJoint2	**65.1**±0.0	**65.1**±0.0	62.0±3.3	**62.6**±2.6
softEM	**65.1**±0.00	57.6±4.6	**65.1**±0.0	57.4±5.2
LG	56.8	0.1	58.5	0.1
NB	**65.1**	60.2	NA	NA
	Heart			
cvxJoint1	**63.0**±6.4	53.3±1.8	63.0±7.4	61.0±6.2
cvxJoint2	55.6±0.0	**55.5**±0.0	**64.0**±7.5	**61.3**±7.1
softEM	55.6±0.0	51.7±1.6	55.6±0.0	52.7±0.0
LG	57.4	0.4	55.2	0.4
NB	55.6	53.0	NA	NA

Table 6.1: Experimental results for the joint generative model. Here cvxJoint1 is cvxJoint followed by SC rounding, whereas cvxJoint2 uses additional re-optimization. Best results in **bold**.

Chapter 7

Conclusion

The main contribution of this thesis is new convex relaxations for clustering with regular Bregman divergences modelling all the probability distributions under regular exponential families. One of the key results is a tighter convex relaxation of hard generative models for Bregman divergence clustering that also accounts for cluster size through *normalized* equivalence relations. In addition, we design efficient new algorithms that optimize the resulting *nonlinear* SDPs based on recent developments in matrix learning techniques. By applying standard rounding methods, we observe that the proposed new convex relaxations for clustering deliver a lower sum of intra-cluster divergences and more faithful alignment with class labels in practice. Finally, applying our formulation to discriminative models immediately leads to *normalized* equivalence relations, which automatically alleviate the problem of imbalanced cluster assignment faced by current relaxations. Additionally, the formulation allows much more efficient optimization.

For future work, it will be interesting to extend these approaches to generative soft clustering. Also, the analysis of approximation gap for these convex relaxations would be of great interest. Since clustering has wide application in real-world problems, it would be also worth further investigation into scaling up the optimization to large applications.

Chapter 8

Appendix

8.1 Tightness of Relaxation of $\mathcal{M}_1$

We show here that $\mathcal{M}_1$ is not the convex hull of $\mathcal{M}$. Our proof is by constructing a new convex relaxation of $\mathrm{conv}\mathcal{M}$ that is a *proper* subset of $\mathcal{M}_1$:

$$\mathcal{M}_{\mathcal{S}} := \{M : \mathbf{0} \preceq M \preceq I, \gamma_{\mathcal{S}}(M) \leq d, M_{i:} \in \Delta_t\},$$

where $\mathcal{S} = \left\{ \frac{1}{\|\mathbf{u}\|^2} \mathbf{u}\mathbf{u}' : \mathbf{u} \in \{0,1\}^t \right\}$, and γ_S is the gauge function of $\mathcal{S}$: $\gamma_{\mathcal{S}}(M) := \inf_{\lambda \geq 0, M \in \lambda \cdot \mathrm{conv}(\mathcal{S})} \lambda$. Clearly $\mathcal{M}_{\mathcal{S}}$ is convex and $\mathcal{M} \subseteq \mathcal{M}_{\mathcal{S}}$. Similarly, $\mathcal{M}_1$ can be rewritten as

$$\mathcal{M}_1 = \{M : \mathbf{0} \preceq M \preceq I, \gamma_{\mathcal{B}}(M) \leq d, M_{i:} \in \Delta_t\},$$

where $\mathcal{B} = \{\mathbf{v}\mathbf{v}' : \|\mathbf{v}\| \leq 1\}$. It is easy to see that $\gamma_{\mathcal{S}}(M) \leq d$ is strictly more restrictive than $\gamma_{\mathcal{B}}(M) \leq d$ because $\mathcal{S} \subsetneq \mathcal{B}$. Therefore it is conceivable that $\mathcal{M}_{\mathcal{S}} \subsetneq \mathcal{M}_1$, and the rest of this appendix section will be devoted to constructing an element in $\mathcal{M}_1 \backslash \mathcal{M}_{\mathcal{S}}$. In essence, $\mathcal{M}_1$ and $\mathcal{M}_{\mathcal{S}}$ employ doubly positive relaxation and completely positive factorization respectively, and their gap has been well studied (Berman & Xu, 2004). Note it is still open as to whether $\mathcal{M}_{\mathcal{S}}$ is the convex hull of $\mathcal{M}$. In terms of optimization, it is much more convenient to use the relaxation $\mathcal{M}_1$ because the $\gamma_{\mathcal{S}}(M)$ term in $\mathcal{M}_{\mathcal{S}}$ is hard to evaluate. In particular the separation oracle is NP-hard: $\max_{Z \in \mathcal{S}} \langle Z, X \rangle$ for a given X.

To construct an element in $\mathcal{M}_1 \backslash \mathcal{M}_{\mathcal{S}}$, we exploit the difference between doubly positive matrices and completely positive matrices. Let $\mathcal{D}_n$ denote the set of $t \times t$ doubly positive matrices, *i.e.* real symmetric matrices that are positive semi-definite and elementwise nonnegative. Let $\mathcal{C}_t$ denote the set of $t \times t$ completely positive matrices, *i.e.* real matrices that can be written as AA', where A is a $t \times k$ elementwise nonnegative matrix ($k \in \mathbb{N}$). It is well known that $\mathcal{C}_t \subsetneq \mathcal{D}_t$ when $t \geq 5$.

Clearly $\mathcal{M}_1$ is the intersection of $\mathcal{D}_t$ with

$$F := \{M : M \preceq I, \mathrm{tr}(M) \leq d, M\mathbf{1} = \mathbf{1}\}.$$

Since $\mathcal{M}_{\mathcal{S}} \subseteq \mathcal{C}_t$, to find $M \in \mathcal{M}_1 \backslash \mathcal{M}_{\mathcal{S}}$ it suffices to find $M \in \mathcal{M}_1$ such that $M \notin \mathcal{C}_t$. Berman & Xu (2004) gave a sufficient and necessary condition for a matrix to be in $\mathcal{D}_5 \backslash \mathcal{C}_5$, under mild assumptions on the structure of the matrix. So we only need to further restrict this condition to F.

Let $t = 5$. Consider a matrix M of the form

$$M = \begin{pmatrix} Y & \boldsymbol{\alpha} & \boldsymbol{\beta} \\ \boldsymbol{\alpha}' & 1 & 0 \\ \boldsymbol{\beta}' & 0 & 1 \end{pmatrix}.$$

Denote the Schur complement as $C = Y - \boldsymbol{\alpha}\boldsymbol{\alpha}' - \boldsymbol{\beta}\boldsymbol{\beta}'$.

Theorem 9. (Berman & Xu, 2004, Theorem 4.2) *Suppose $Y \in \mathcal{D}_3$, $M \in \mathcal{D}_5$, and $\mathrm{rank}(M) = 3$. Then $M \in \mathcal{D}_5 \backslash \mathcal{C}_5$ if and only if*

- *There are exactly two negative components above the diagonal in C, and*
- $\lambda_4 + \lambda_5 < 1$, *where*

$$\lambda_4 = \min_{1 \leq i < j \leq 3} \left\{ \frac{\alpha_i \alpha_j}{-C_{ij}} \middle| C_{ij} < 0 \right\},$$
$$\lambda_5 = \min_{1 \leq i < j \leq 3} \left\{ \frac{\beta_i \beta_j}{-C_{ij}} \middle| C_{ij} < 0 \right\}.$$

Since d is a parameter, it can be set in our favor and so we ignore it for now. Also we can scale F by

$$F_\rho := \{M : M \preceq (\rho + 1)I, M\mathbf{1} = (\rho + 1)\mathbf{1}\},$$

where $\rho > 0$ is a constant. So it suffices to find $M \in \mathcal{D}_5 \cap F_\rho$ such that $M \notin \mathcal{C}_5$, *i.e.* $M \in (\mathcal{D}_5 \backslash \mathcal{C}_5) \cap F_\rho$. Now let us apply Theorem 9.

1. Since $\mathrm{rank}(M) = \mathrm{rank}(C) + 2 = 3$ (property of Schur complement), we can assume $C = \boldsymbol{\gamma}\boldsymbol{\gamma}'$. So

$$Y = \boldsymbol{\alpha}\boldsymbol{\alpha}' + \boldsymbol{\beta}\boldsymbol{\beta}' + \boldsymbol{\gamma}\boldsymbol{\gamma}'. \tag{8.1}$$

2. Since $M\mathbf{1} = (\rho+1)\mathbf{1}$, we have $\boldsymbol{\alpha}'\mathbf{1} = \boldsymbol{\beta}'\mathbf{1} = \rho$, and

$$\begin{aligned} & Y\mathbf{1} + \boldsymbol{\alpha} + \boldsymbol{\beta} = (\rho+1)\mathbf{1} \\ \Leftrightarrow\ & (\boldsymbol{\alpha}\boldsymbol{\alpha}' + \boldsymbol{\beta}\boldsymbol{\beta}' + \boldsymbol{\gamma}\boldsymbol{\gamma}')\mathbf{1} + \boldsymbol{\alpha} + \boldsymbol{\beta} = (\rho+1)\mathbf{1} \\ \Leftrightarrow\ & \boldsymbol{\gamma}\boldsymbol{\gamma}'\mathbf{1} + (\rho+1)(\boldsymbol{\alpha}+\boldsymbol{\beta}) = (\rho+1)\mathbf{1}. \end{aligned} \tag{8.2}$$

Left multiply it by $\mathbf{1}'$, we obtain

$$(\boldsymbol{\gamma}'\mathbf{1})^2 + 2(\rho+1)\rho = 3(\rho+1). \tag{8.3}$$

So we first randomly generate $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ that are elementwise nonnegative and $\boldsymbol{\alpha}'\mathbf{1} = \boldsymbol{\beta}'\mathbf{1} = \rho$. Then $\boldsymbol{\gamma}$ can be determined by using (8.2) and (8.3) (up to negation).

By (8.3), we must set $\rho < 1.5$.

3. Check if $C = \boldsymbol{\gamma}\boldsymbol{\gamma}'$ has exactly two negative components above its diagonal. If not, then regenerate $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

4. Check if $\lambda_4 + \lambda_5 < 1$ and Y from (8.1) is elementwise nonnegative ($Y \succeq \mathbf{0}$ is guaranteed by construction). If not, then regenerate $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

5. Check if the maximum eigenvalue of M is $\rho + 1$. If not, regenerate $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

6. Scale M down by multiplying it with $1/(\rho+1)$. Set

$$\begin{aligned} d &= (\mathrm{tr}(Y) + 2)/(1+\rho) \\ &= (\|\boldsymbol{\alpha}\|^2 + \|\boldsymbol{\beta}\|^2 + \|\boldsymbol{\gamma}\|^2 + 2)/(1+\rho). \end{aligned}$$

In our experiments, we set $\rho = 1.25$ and found an example matrix immediately.

8.2 Characterizing $\Omega(T)$

8.2.1 Ω is a norm

Note that $\Omega(T)$ depends only on the singular values of T. So it suffices to show that $\kappa(\mathbf{s}) := \sqrt{f(\mathbf{s})}$ is a symmetric gauge (Horn & Johnson, 1985, Theorem 3.5.18), where $f(\mathbf{s})$ is defined in (4.10). Clearly $\kappa(\mathbf{s})$ is permutation invariant, $\kappa(a\mathbf{s}) = |a|\,\kappa(\mathbf{s})$ for all $a \in \mathbb{R}$, and $\kappa(\mathbf{s}) = 0$ iff $\mathbf{s} = \mathbf{0}$. So it suffices to prove the triangle inequality for $\kappa(\mathbf{s})$. For any $\mathbf{s}_1$ and $\mathbf{s}_2$, let $t_1 = \kappa(\mathbf{s}_1)$ and $t_2 = \kappa(\mathbf{s}_2)$. Then $\kappa(\frac{\mathbf{s}_1}{t_1}) = \kappa(\frac{\mathbf{s}_2}{t_2}) = 1$, and

$$\frac{\mathbf{s}_1+\mathbf{s}_2}{t_1+t_2}=\frac{t_1}{t_1+t_2}\frac{\mathbf{s}_1}{t_1}+\frac{t_2}{t_1+t_2}\frac{\mathbf{s}_2}{t_2}. \tag{8.4}$$

Note $f(\mathbf{s})$ is convex because $\sum_i s_i^2/\sigma_i$ is jointly convex in $(\mathbf{s},\boldsymbol{\sigma})$, and $f(\mathbf{s})$ just minimizes out $\boldsymbol{\sigma}$. So the sub-level set at level 1 for f (and κ) is convex. Therefore by (8.4), $\kappa((\mathbf{s}_1+\mathbf{s}_2)/(t_1+t_2))\leq 1$, and so $\kappa(\mathbf{s}_1+\mathbf{s}_2)\leq t_1+t_2=\kappa(\mathbf{s}_1)+\kappa(\mathbf{s}_2)$. The claim follows.

8.2.2 The dual norm of Ω

Given a matrix R, the dual norm of Ω is defined by

$$\Omega_*(R)=\max_{T:\Omega(T)\leq 1}\operatorname{tr}(R'T). \tag{8.5}$$

Let the SVD of R be $R=U\operatorname{diag}\{r_1,\ldots,r_t\}V'$, where $r_1\geq\ldots\geq r_t$. Since Ω is defined via the singular values of T, again by von Neumann's trace inequality the maximum is attained when the left and right singular values of T are U and V, respectively. Then

$$\Omega_*(R)=\max_{\mathbf{s}:f(\mathbf{s})\leq 1}\mathbf{r}'\mathbf{s}, \tag{8.6}$$

which by (4.10) is equivalent to

$$\begin{gathered}\max_{\mathbf{s},\boldsymbol{\sigma}}\mathbf{r}'\mathbf{s},\\ \text{subject to } \sigma_i\in[0,1],\ \textstyle\sum_{i=1}^t\sigma_i\leq d-1,\ \sum_{i=1}^t\frac{s_i^2}{\sigma_i}\leq 1.\end{gathered} \tag{8.7}$$

Using the Cauchy-Schwarz inequality, we have

$$\begin{aligned}\mathbf{r}'\mathbf{s}=\sum_{i=1}^t\frac{s_i}{\sqrt{\sigma_i}}\cdot r_i\sqrt{\sigma_i}&\leq\left(\sum_{i=1}^t\frac{s_i^2}{\sigma_i}\right)^{1/2}\left(\sum_{i=1}^t r_i^2\sigma_i\right)^{1/2}\\ &\leq\left(\sum_{i=1}^t r_i^2\sigma_i\right)^{1/2}\leq\left\|(r_1,r_2,\ldots,r_{d-1})'\right\|,\end{aligned} \tag{8.8}$$

where the last two inequalities use the constraints in (8.7). The equalities can all be attained by setting $s_i=r_i/\left\|(r_1,r_2,\ldots,r_{d-1})'\right\|$ and $\sigma_i=1$ for $i\leq d-1$, and $s_i=0$ and $\sigma_i=0$ for $i\geq d$. Clearly $U\operatorname{diag}(\mathbf{s})V'$ is a subgradient of Ω_* at R. Evaluating the dual norm is inexpensive, since it requires only the top $d-1$ singular values of R.

On the basis of the above discussion, we will then extend our strategy to $\mathcal{M}_2$ in the following section.

8.3 Characterizing $\Xi(T)$

8.3.1 $\Xi(T)$ is a norm

Based on (4.21), it is quite easy to see that $\Xi(T)$ is a norm. Trivially, $\Xi(aT) = |a|\,\Xi(T)$ for all $a \in \mathbb{R}$. To make $\Xi(T) = 0$, we need $\Omega(HT) = 0$ and $\|T'\mathbf{1}\| = 0$. Since Ω is a norm, so we need $HT = \mathbf{0}$ and $T'\mathbf{1} = \mathbf{0}$. Therefore $T = IT = (H + \frac{1}{t}\mathbf{1}\mathbf{1}')T = \mathbf{0}$. Finally, since both $\Omega(HT)$ and $\frac{1}{\sqrt{t}}\|T'\mathbf{1}\|$ are semi-norms in T, it is easy to verify that $\Xi(T)$ also satisfies the triangle inequality.

8.3.2 The dual norm of $\Xi(T)$

Given G, the dual norm of $\Xi(\cdot)$ on G is

$$\begin{aligned}
&\Xi_*(G) = \max_{T:\Xi(T)\le 1} \operatorname{tr}(G'T) \\
=& \max_{T:\Omega^2(HT)+\frac{1}{t}\|T'\mathbf{1}\|^2\le 1} \operatorname{tr}(G'T) \\
=& \max_{T:\Omega^2(HT)+\frac{1}{t}\|T'\mathbf{1}\|^2\le 1} \operatorname{tr}((HG + \frac{1}{t}\mathbf{1}\mathbf{1}'G)'(HT + \frac{1}{t}\mathbf{1}\mathbf{1}'T)) \\
=& \max_{T:\Omega^2(HT)+\frac{1}{t}\|T'\mathbf{1}\|^2\le 1} \operatorname{tr}((HG)'(HT)) + \frac{1}{t}(G'\mathbf{1})'(T'\mathbf{1}).
\end{aligned}$$

We can optimize HT and $T'\mathbf{1}$ *independently* because

Proposition 10. $\{(HT, T'\mathbf{1}) : T\} = \{(S, \mathbf{v}) : S'\mathbf{1} = \mathbf{0}\}$.

Proof. $\subseteq$ is obvious because $(HT)'\mathbf{1} = \mathbf{0}$. For $\supseteq$, just define $T = S + \frac{1}{t}\mathbf{1}\mathbf{v}'$. Then $HT = HS = HS + \frac{1}{t}\mathbf{1}\mathbf{1}'S = S$ and $T'\mathbf{1} = S'\mathbf{1} + \frac{1}{t}\mathbf{v}\mathbf{1}'\mathbf{1} = \mathbf{v}$. □

By Proposition 10, the problem becomes

$$\max_{S,\mathbf{v}:S'\mathbf{1}=\mathbf{0},\Omega^2(S)+\frac{1}{t}\|\mathbf{v}\|^2\le 1} \operatorname{tr}((HG)'S) + \frac{1}{t}(G'\mathbf{1})'\mathbf{v}.$$

Denote $\|\mathbf{v}\| = \tau$, then $(G'\mathbf{1})'\mathbf{v} \le \tau\,\|G'\mathbf{1}\|$ with equality attained at $\mathbf{v} = \tau G'\mathbf{1}/\,\|G'\mathbf{1}\|$. So the problem can be further reformulated as

$$\max_{\tau\in[0,\sqrt{t}]} \max_{S:S'\mathbf{1}=\mathbf{0},\Omega^2(S)\le 1-\frac{\tau^2}{t}} \operatorname{tr}((HG)'S) + \frac{\tau}{t}\left\|G'\mathbf{1}\right\|.$$

In the inner optimization over S, if we ignore the $S'\mathbf{1} = \mathbf{0}$ constraint, then by the discussion on how to compute Ω_* in Section 4.2, the left and right singular vectors of the optimal S are the same as

those of HG. Since $(HG)'\mathbf{1} = \mathbf{0}$, so $S'\mathbf{1} = \mathbf{0}$ is automatically satisfied. Then the problem becomes

$$\begin{aligned}
\Xi_*(G) &= \max_{\tau\in[0,\sqrt{t}]} \left\{ \frac{\tau}{t}\left\|G'\mathbf{1}\right\| + \max_{S:\Omega_*(S)\le\sqrt{1-\frac{\tau^2}{t}}} \operatorname{tr}((HG)'S) \right\} \\
&= \max_{\tau\in[0,\sqrt{t}]} \frac{\tau}{t}\left\|G'\mathbf{1}\right\| + \Omega(HG)\sqrt{1-\frac{\tau^2}{t}} \\
&= \max_{\tau\in[0,\sqrt{t}]} \frac{1}{\sqrt{t}}\left\|G'\mathbf{1}\right\| \frac{\tau}{\sqrt{t}} + \Omega(HG)\sqrt{1-\frac{\tau^2}{t}} \\
&= \left(\frac{1}{t}\left\|G'\mathbf{1}\right\|^2 + \Omega^2(HG)\right)^{\frac{1}{2}} \left(\frac{\tau^2}{t} + 1 - \frac{\tau^2}{t}\right)^{\frac{1}{2}} \qquad (8.9) \\
&= \sqrt{\frac{1}{t}\left\|G'\mathbf{1}\right\|^2 + \Omega^2(HG)},
\end{aligned}$$

where (8.9) uses Cauchy-Schwartz and the optimal τ is attained at

$$\tau^* = \frac{\left\|G'\mathbf{1}\right\|\sqrt{t}}{\sqrt{\left\|G'\mathbf{1}\right\|^2 + t\Omega^2(HG)}} \ (< \sqrt{t}).$$

The optimal T is

$$T^* = \sqrt{1-\frac{(\tau^*)^2}{t}} \operatorname*{argmax}_{S:\Omega_*(S)\le 1} \operatorname{tr}((HG)'S) + \frac{\tau^*}{t\left\|G'\mathbf{1}\right\|}\mathbf{1}\mathbf{1}'G.$$

Again, this procedure only requires the top $d-1$ singular values of HG.

Bibliography

Aloise, D., Seshpande, A., Hansen, P., and Popat, P. Np-hardness of Euclidean sum-of-squares clustering. *Machine Learning*, 75:245–249, 2009.

Anandkumar, A., Hsu, D., and Kakade, S. A method of moments for mixture models and hidden Markov models. In *Proc. Conference on Learning Theory*, 2012.

Arora, S. and Kannan, R. Learning mixtures of separated nonspherical Gaussians. *The Annals of Applied Probability*, 15(1A):69–92, 2005.

Arthur, D. and Vassilvitskii, S. k-means++: The advantage of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2007.

Bach, F. and Harchaoui, Z. Diffrac: A discriminative and flexible framework for clustering. In *Advances in Neural Information Processing Systems 20*, 2007.

Banerjee, A., Merugu, S., Dhillon, I. S., and Ghosh, J. Clustering with Bregman divergences. *Journal of Machine Learning Research*, 6:1705–1749, 2005.

Berman, A. and Xu, C. 5×5 completely positive matrices. *Linear Algebra and its Applications*, 393:55–71, 2004.

Borwein, J. M. and Lewis, A. S. *Convex Analysis and Nonlinear Optimization: Theory and Examples*. CMS books in Mathematics. Canadian Mathematical Society, 2000.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–123, 2010.

Chapelle, O., Schölkopf, B., and Zien, A. (eds.). *Semi-Supervised Learning*. MIT Press, 2006.

Chaudhuri, K., Dasgupta, S., and Vattani, A. Learning mixtures of Gaussians using the k-means algorithm. arXiv:0912.0086v1, 2009.

Cheng, H., Zhang, X., and Schuurmans, D. Convex relaxations of bregman divergence clustering. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence*, 2013.

Dasgupta, S. The hardness of k-means clustering. Technical Report CS2008-0916, CSE Department, UCSD, 2008.

Dasgupta, S. and Schulman, L. A probabilistic analysis of EM for mixtures of separated, spherical Gaussians. *Journal of Machine Learning Research*, 8:203–226, 2007.

Dempster, A., Laird, N., and Rubin, D. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39(1):1–22, 1977.

Frank, A. and Asuncion, A. UCI machine learning repository, 2010. URL `http://archive.ics.uci.edu/ml`.

Guo, Y. and Schuurmans, D. Convex relaxations of latent variable training. In *Advances in Neural Information Processing Systems 20*, 2007.

Hansen, P., Jaumard, B., and Mladenovic, N. Minimum sum of squares clustering in a low dimensional space. *Journal of Classification*, 15(1):37–55, 1998.

Horn, R. and Johnson, C. *Matrix Analysis*. Cambridge University Press, Cambridge, 1985.

Hsu, D. and Kakade, S. Learning mixtures of spherical Gaussians: Moment methods and spectral decompositions. In *Innovations in Theoretical Computer Science (ITCS)*, 2013.

Inaba, M., Katoh, N., and Imai, H. Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering. In *Proc. Symp. Computational Geometry*, 1994.

Joulin, A. and Bach, F. A convex relaxation for weakly supervised classifiers. In *Proceedings of the International Conference on Machine Learning*, 2012.

Joulin, A., Bach, F., and Ponce, J. Efficient optimization for discriminative latent class models. In *Advances in Neural Information Processing Systems 23*, 2010.

K. Bredies, D. Lorenz and Maass, P. A generalized conditional gradient method and its connection to an iterative shrinkage method. *Computational Optimization and Applications*, 42:173–193, 2009. ISSN 0926-6003. doi: 10.1007/s10589-007-9083-3. URL `http://dx.doi.org/10.1007/s10589-007-9083-3`.

Kalai, A., Moitra, A., and Valiant, G. Efficiently learning mixtures of two Gaussians. In *Proceedings ACM Symposium on Theory of Computing*, 2010.

Kumar, A., Sabharwal, Y., and Sen, S. A simple linear time $(1+\epsilon)$-approximation algorithm for k-means clustering in any dimensions. In *Proc. Symposium on Foundations of Computer Science,*, 2004.

Lashkari, D. and Golland, P. Convex clustering with exemplar-based models. In *Advances in Neural Information Processing Systems 20*, 2007.

Laue, S. A hybrid algorithm for convex semidefinite optimization. In *Proceedings of the International Conference on Machine Learning*, 2012.

MacQueen, J. Some methods of classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281. 1967.

Mirsky, L. *An Introduction to Linear Algebra*. Oxford, 1955.

Mirsky, L. A trace inequality of John von Neumann. *Monatsh. Math.*, 79(4):303–306, 1975.

Moitra, A. and Valiant, G. Settling the polynomial learnability of mixtures of Gaussians. In *Proc. Symposium on Foundations of Computer Science,*, 2010.

Neal, R. and Hinton, G. A view of the EM algorithm that justifies incremental, sparse, and other variants. In Jordan, M. (ed.), *Learning in Graphical Models*. Kluwer, 1998.

Ng, A., Jordan, M., and Weiss, Y. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, 2001.

Nowozin, S. and Bakir, G. A decoupled approach to exemplar-based unsupervised learning. In *Proceedings of the International Conference on Machine Learning*, 2008.

Peng, J. and Wei, Y. Approximating k-means-type clustering via semidefinite programming. *SIAM J. on Optimization*, 18:186–205, 2007.

Shi, J. and Malik, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

Tsuda, K., Rätsch, G., and Warmuth, M. Matrix exponentiated gradient updates for on-line learning and Bregman projections. In *Advances in Neural Information Processing Systems 17*, 2004.

Wang, S. and Schuurmans, D. Learning continuous latent variable models with Bregman divergence. In *International Conference on Algorithmic Learning Theory*, 2003.

Xing, E. and Jordan, M. On semidefinite relaxation for normalized k-cut and connections to spectral clustering. Technical Report UCB/CSD-03-1265, EECS Department, University of California, Berkeley, 2003.

Xu, L. and Schuurmans, D. Unsupervised and semi-supervised multi-class support vector machines. In *Proc. Conf. Association for the Advancement of Artificial Intelligence (AAAI)*, 2005.

Zass, R. and Shashua, A. A unifying approach to hard and probabilistic clustering. In *Proc. Intl. Conf. Computer Vision*, 2005.

Zha, H., Ding, C., Gu, M., He, X., and Simon, H. Spectral relaxation for k-means clustering. In *Advances in Neural Information Processing Systems (NIPS)*, 2001.

Zhang, X., Yu, Y., and Schuurmans, D. Accelerated training for matrix-norm regularization: A boosting approach. In *Advances in Neural Information Processing Systems 25*, 2012.