Information Theoretical Clustering via Semidefinite Programming

Meihong Wang
Department of Computer Science
U. of Southern California
Los Angeles, CA 90089
meihongw@usc.edu

Fei Sha
Department of Computer Science
U. of Southern California
Los Angeles, CA 90089
feisha@usc.edu

Abstract

We propose techniques of convex optimization for information theoretical clustering. The clustering objective is to maximize the mutual information between data points and cluster assignments. We formulate this problem first as an instance of MAX K CUT on weighted graphs. We then apply the technique of semidefinite programming (SDP) relaxation to obtain a convex SDP problem. We show how the solution of the SDP problem can be further improved with a low-rank refinement heuristic. The low-rank solution reveals more clearly the cluster structure of the data. Empirical studies on several datasets demonstrate the effectiveness of our approach. In particular, the approach outperforms several other clustering algorithms when compared on standard evaluation metrics.

1 INTRODUCTION

Clustering is an important problem in machine learning and data mining. The basic setup is to group data points into disjoint partitions that optimize some criteria. For instance, the technique of K-means minimizes the sum of pairwise distances between data points in the same partition. The algorithm iterates between two steps: computing the centroids of existing partitions and re-assigning every data point to the partition with the nearest centroid. Both steps monotonically decrease the optimality measure, and the algorithm converges to a local optimum.

Appearing in Proceedings of the 14^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2011, Fort Lauderdale, FL, USA. Volume 15 of JMLR: W&CP 15.

Many variants to K-means exist. If data lie on a low-dimensional submanifold, then we can use (geodesic) distances on the manifold instead of Euclidean distances in the embedding space. This leads to the technique of spectral clustering (Ng et al., 2001).It is also easy to see how kernel tricks can be applied to formulate distances with inner products in nonlinear feature spaces, resulting kernelized K-means.

Information theoretic clustering (ITC) has recently been investigated by Faivishevsky and Goldberger (2010) as an alternative criterion. The criterion maximizes the mutual information (MI) between data points and their cluster memberships. To overcome the difficulty of estimating MI between high-dimensional variables, ITC uses pairwise distances based non-parametric statistics (Wang et al., 2009; Kozachenko and Leonenko, 1987). Maximizing the mutual information criterion, however, still remains challenging as it is a NP-hard combinator optimization. The earlier work uses a local search procedure, sequentially and greedily re-assigning a data point from its current cluster to a new one. While seemingly effective, there is no established theoretical properties on how well this procedure can achieve.

In this paper, we propose a new optimization procedure for ITC. We first identify the problem as an instance of MAX K CUT on weights graphs. We then apply semidefinite programming (SDP) relaxation to find approximate solutions. The relaxed problem is convex and can be solved efficiently. Furthermore, the SDP-based solutions have a strong theoretical guarantee in approximation factors. Empirical studies also show that our approach yields much higher clustering quality than the heuristic procedure.

The rest of the paper is organized as follows. In section 2, we describe nonparametric techniques of estimating entropy and mutual information. In section 3, we describe the idea of information theoretical clustering and the process of relaxing it as an instance of SDP. We discuss related work in section 4. Experimen-

tal results are presented in section 5. We summarize in section 6 and discuss future directions for research.

2 ESTIMATION OF ENTROPY

Entropy plays an important role in forming many information theoretical quantities. In this paper, we are interested in how it is related to $I(X;Y)$, the mutual information between a random variable X and its cluster membership Y. Concretely,

$$I(X;Y) = H(X) - H(X|Y), \tag{1}$$

where the right-hand-side is the difference between the entropy and the conditional entropy of X.

For high-dimensional X, estimating entropies from samples is a challenging problem. Standard approaches include quantizing/binning X or constructing density estimators of X. They often do not work well due to the "curse of dimensionality", where the number of data points grows exponentially in order to obtain an accurate estimation.

There has been a growing interest in using nonparameteric statistics to estimate entropies. Specifically, it was shown that, given N samples $\mathcal{D} = \{\boldsymbol{x}_1, ...\boldsymbol{x}_\mathsf{N}\}$ where $\boldsymbol{x}_i \in \mathbb{R}^\mathsf{D}$, the entropy of X can be estimated by

$$\hat{H}_k(X) = \frac{\mathsf{D}}{\mathsf{N}} \sum_i \log \|\boldsymbol{x}_i - \boldsymbol{x}_i^{(k)}\|_2^2 + \mathsf{const}, \tag{2}$$

where $\boldsymbol{x}_i^{(k)}$ is the k-th nearest neighbor of $\boldsymbol{x}_i$ in $\mathcal{D}$ (Wang et al., 2009; Kozachenko and Leonenko, 1987). The estimator approaches $H(X)$ with a convergence rate of $O(1/\sqrt{\mathsf{N}})$.

Averaging $\hat{H}_k(X)$ over all possible k from 1 to $(\mathsf{N}-1)$ leads to a simplified estimator,

$$\begin{aligned}\hat{H}(X) &= \frac{1}{(\mathsf{N}-1)} \sum_{k=1}^{N-1} \hat{H}_k(X) \\ &= \frac{\mathsf{D}}{\mathsf{N}(\mathsf{N}-1)} \sum_{i \neq j} \log \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 + \mathsf{const}. \end{aligned} \tag{3}$$

This estimator was first investigated in (Faivishevsky and Goldberger, 2009, 2010) and can be understood intuitively as follows.

To estimate the entropy, one would need to obtain an unbiased estimator of $-\log p(\boldsymbol{x}_i)$ such that

$$H(X) \approx \frac{1}{\mathsf{N}} \sum_i -\log p(\boldsymbol{x}_i). \tag{4}$$

For one-dimensional X, we can approximate $p(\boldsymbol{x}_i)$ as a uniform distribution between $\boldsymbol{x}$ and $\boldsymbol{x}_j$, which gives rise to

$$-\log p(x_i) \approx -\log \frac{1}{|x_i - x_j|}. \tag{5}$$

Averaging this estimator over all possible $x_j \neq x_i$, we obtain an estimator in the form of eq. (3). A detailed derivation of this result is given in (Faivishevsky and Goldberger, 2010).

$\hat{H}(X)$ is more computationally convenient than $\hat{H}_k(X)$ as it does not need to identify nearest neighbors. Thus, we focus on $\hat{H}(X)$ in the rest of the paper.

For the conditional entropy $H(X|Y)$, we estimate it with the following

$$\hat{H}(X|Y) = \sum_y \hat{p}(Y=y)\hat{H}(X, Y=y), \tag{6}$$

where $\hat{p}(Y=y)$ is the (empirical) prior distribution and $\hat{H}(X, Y=y)$ is the entropy of data samples whose corresponding Y is y.

Specifically, in the context of clustering, Y stands for cluster memberships. We assume that there are K clusters, each with N_k data points. The conditional entropy is thus given by (up to a constant)

$$\begin{aligned}\hat{H}(X|Y) &= \sum_{k=1}^{\mathsf{K}} \frac{\mathsf{N}_k}{\mathsf{N}} \frac{\mathsf{D}}{\mathsf{N}_k(\mathsf{N}_k-1)} \sum_{Y_i=Y_j=k} \log \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 \\ &= \sum_{k=1}^{\mathsf{K}} \frac{\mathsf{D}}{\mathsf{N}(\mathsf{N}_k-1)} \sum_{Y_i=Y_j=k} \log \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 \end{aligned} \tag{7}$$

where the inner summation is over data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ which are both assigned to the cluster k.

Since the entropy $H(X)$ does not depend on how we assign data points to different clusters, maximizing the mutual information – a clustering criterion to be described in detail in the next section – is equivalent to minimizing the conditional entropy. We gain further insight by contrasting the conditional entropy to the criterion minimized in the K-means:

$$\begin{aligned}J(X,Y) &= \sum_{k=1}^{\mathsf{K}} \sum_{Y_i=k} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|_2^2 \\ &= \sum_{k=1}^{\mathsf{K}} \sum_{Y_i=Y_j=k} \frac{1}{\mathsf{N}_k} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 \end{aligned} \tag{8}$$

where $\boldsymbol{\mu}_k$ is the centroid of the cluster k. Both $H(Y|X)$ and $J(X,Y)$ measure how tight the clusters are. However, the conditional entropy uses the logarithm of the distances, which grows slower than the linear function used by the K-means. Thus, arguably the conditional entropy tends to be less sensitive and more robust to outliers. In the following, we describe how

we can minimize the conditional entropy and therefore, obtain an optimal clustering in the information theoretical sense.

3 INFORMATION THEORETICAL CLUSTERING

The conditional entropy of eq. (7) depends on the cluster memberships of the data points. The minimization of this quantity over all possible assignments is referred as information theoretical clustering (ITC) in (Faivishevsky and Goldberger, 2010). Experimental results reported there have shown this is an effective and useful clustering criterion.

Despite its similarity to the K-means objective function in eq. (8), minimizing the conditional entropy does not admit the two-step alternate minimization procedure often used in K-means. Specifically, it is not obvious how to define a single centroid (as in K-means) for each cluster and iteratively update the locations of these centroids.

Instead, Faivishevsky and Goldberger (2010) proposed a local search procedure that greedily assigns data points to clusters. The procedure starts with a random assignment. Then a data point is cyclically but randomly chosen from $\mathcal{D}$ and evaluated. If changing its cluster membership k to a different k' would result in reduction in the conditional entropy, the data point's cluster membership will be updated to k'. It is easy to see that the procedure converges to a local optimum. Furthermore, determining whether to change assignments can be performed efficiently, involving at most $(\mathsf{N}-1)$ calculations of distances. However, there is no rigorous analysis on how many such evaluations are needed in order to converge or how good the converged solution is in terms of approximation guarantee.

In what follows, we show how we can relax the minimization problem to an instance of semidefinite programming, which is solvable efficiently and provides provably approximation performance guarantee.

3.1 Integer programming formulation

Let $a_{ki} \in \{0,1\}$ denote whether $\boldsymbol{x}_i$ is assigned to the cluster k. These indicator variables form naturally an assign matrix $\boldsymbol{A} \in \{0,1\}^{\mathsf{K}\times\mathsf{N}}$. Moreover, let $\boldsymbol{a}_i$ denote the i-th column of $\boldsymbol{A}$, ie, the assignment vector for $\boldsymbol{x}_i$.

Our first simplification is to assume that each cluster has an equal number of data points, i.e., $\mathsf{N}_k = N/\mathsf{K}$. We discuss later the effect of this assumption. Under this assumption, the conditional entropy is given by,

$$\begin{aligned}\hat{H}(X|Y) &= \sum_k \frac{1}{\mathsf{N}_k - 1} \sum_{i\neq j} a_{ki}a_{kj} \log \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 \\ &= \sum_{ij} \boldsymbol{a}_i{}^\top \boldsymbol{a}_j \log \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 \\ &= \sum_{ij} G_{ij} L_{ij} = \mathrm{Trace}[\boldsymbol{G}\boldsymbol{L}] \end{aligned} \tag{9}$$

where G_{ij} stands for the (i,j)-th element of the Gram matrix $\boldsymbol{G} = \boldsymbol{A}^\top\boldsymbol{A}$. $L_{ij} = \log \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2$ denotes the (i,j)-th element of matrix $\boldsymbol{L}$.

Thus, we have formulated the problem of informational theoretical clustering (ITC) in the following integer programming problem

$$\begin{aligned} (\mathcal{I}.\mathcal{P}.) \quad & \min \ \mathrm{Trace}[\boldsymbol{G}\boldsymbol{L}] \\ & \text{s.t. } \boldsymbol{G} = \boldsymbol{A}^\top\boldsymbol{A}, \ \ \boldsymbol{A} \in \{0,1\}^{\mathsf{K}\times\mathsf{N}} \\ & \quad \boldsymbol{A}\mathbf{1} = \mathsf{N}/\mathsf{K}\,\mathbf{1} \\ & \quad \mathbf{1}^\top\boldsymbol{A} = \mathbf{1}^\top \end{aligned} \tag{10}$$

where $\mathbf{1}$ denotes the vector whose elements are all ones. The last two constraints state that each cluster has N/K data points and then each data point needs to be assigned to a cluster.

The integer programming problem is NP-hard to solve. In fact, it is an instance of MAX K CUT on weighted graphs (Vazirani, 2001; Hochbaum, 1997). In MAX K CUT, we seek K disjoint partitions that maximize the sum of the weights on the edges which have two vertices in different partitions. It is easy to see that the integer programming implements such cut with pairwise weights given by $-\boldsymbol{L}$. MAX K CUT has recently been attacked with semidefinite programming (SDP) relaxation with great success (Goemans and Williamson, 1995; Frieze and Jerrum, 1997). We adopt the same strategy here.

3.2 SDP relaxation

We first relax the constraint that $\boldsymbol{G}$ needs to be a binary matrix. Instead, we constrain $\boldsymbol{G}$ to be a positive semidefinite matrix whose elements are between 0 and 1. The diagonal elements are constrained to be 1 (as they store the value of $\boldsymbol{a}_i{}^\top\boldsymbol{a}_i$). To eliminate the constraints on the row and column sums of $\boldsymbol{A}$, we note that

$$\boldsymbol{G}\mathbf{1} = \boldsymbol{A}^\top\boldsymbol{A}\mathbf{1} = \boldsymbol{A}\mathsf{N}/\mathsf{K}\mathbf{1} = \mathsf{N}/\mathsf{K}\,\mathbf{1} \tag{11}$$

This gives rise to the following optimization problem

$$\begin{aligned} (\mathcal{SDP}) \quad & \min \ \ \mathrm{Trace}[\boldsymbol{G}\boldsymbol{L}] \\ & \text{s.t.} \ \ 0 \le G_{ij} \le 1, G_{ii} = 1 \\ & \quad \boldsymbol{G}\,\mathbf{1} = \mathsf{N}/\mathsf{K}\,\mathbf{1} \\ & \quad \boldsymbol{G} \succeq 0 \end{aligned} \tag{12}$$

where the last condition $\boldsymbol{G} \succeq 0$ constrains $\boldsymbol{G}$ to be positive semidefinite. This optimization is an instance of semidefinite programming, a convex optimization that can be solved efficiently (Boyd and Vandenberghe, 2004). We have used an off-shelf solver CSDP to solve it (Borchers and Young, 2007).

The constraint $G_{ij} \leq 1$ is redundant. $G_{ii} = 1$ for any i implies that all points of $\boldsymbol{a}_i$ live on the unit-sphere. Thus, G_{ij}, the inner product between those points, are automatically constrained to be at most one.

3.3 Recovering Binary Solution

The solution $\boldsymbol{G}$ from the SDP relaxation eq. (12) approximates the binary matrix $\boldsymbol{A}^\top \boldsymbol{A}$. There are several ways to recover the binary assignment matrix $\boldsymbol{A}$. To this end, Frieze and Jerrum proposed the following randomized algorithm ,

- Obtain the top K eigenvalues and eigenvectors from $\boldsymbol{G}$. Let them be $\{(\lambda_k, \boldsymbol{v}_k)_{k=1}^{\mathsf{K}}\}$.
- Each data point $\boldsymbol{x}_i$ is assigned a K-dimensional coordinate $\hat{\boldsymbol{a}}_i = [\sqrt{\lambda_1}\boldsymbol{v}_{1i}, \sqrt{\lambda_2}\boldsymbol{v}_{2i}, \cdots, \sqrt{\lambda_{\mathsf{K}}}\boldsymbol{v}_{\mathsf{K}i}]^\top$
- Randomly sample K points $\{\boldsymbol{z}_k\}$ on the unit sphere in the K-dimensional space
- Assign $\boldsymbol{x}_i$ to the cluster $y_i = \arg\min_k \|\hat{\boldsymbol{a}}_i - \boldsymbol{z}_k\|_2^2$

Intuitively, the first two steps of their algorithm compute the embedding of all data points in K-dimensional subspace with multidimensional scaling (MDS). The last steps are similar to K-means without iterations, namely, there are no updates to centrods $\boldsymbol{z}_k$ once assignments are completed. Alternatively, these steps can be seen as projecting $\hat{\boldsymbol{a}}_i$ onto random hyperplanes with normal vectors of $\boldsymbol{z}_k$ and picking the cluster leading to the most distant projection from the origin.

It is possible to derandomize the algorithm to obtain deterministic cluster assignments (Engebretsen et al., 2002). In this paper, we take the simpler approach of clustering $\hat{\boldsymbol{a}}_i$ with multiple random restarts.

3.4 Low-rank refinement

One of the most important constraints that we have relaxed from the $\mathcal{I}.\mathcal{P}.$ formulation eq. (10) to the SDP formulation eq. (12) is the elimination of the requirement on the rank of $\boldsymbol{G}$. Therefore, the SDP solution $\boldsymbol{G}$ is likely to be ranked more than K, the number of clusters. Unfortunately, equality constraints on matrix ranks are nonconvex. One common heuristic is to minimize the trace of $\boldsymbol{G}$ since it is the convex envelope of rank constrained sets. This heuristic turns out to be ineffective for eq. (12) as the tract of $\boldsymbol{G}$ is a constant due to the constraints $G_{ii} = 1$.

To add this problem, we use the log*det* heuristic developed by Fazel et al. (2003). Specifically, we minimize $\log\det(\boldsymbol{G} + \epsilon\boldsymbol{I})$ as a *smooth surrogate* for the rank of $\boldsymbol{G}$. $\epsilon > 0$ is a regularizer constant, taking very small values. It is instructive to consider the simple case where $\boldsymbol{G}$ is a scalar g. Thus, the positive semidefinite constraint on $\boldsymbol{G}$ reduces to the constraint $g \geq 0$.

The log*det* heuristic becomes $\log(g + \epsilon)$ and is minimized when $g = 0$, attaining the minimum of the rank of a scalar, ie, zero. For a non-scalar $\boldsymbol{G}$, the heuristic computes $\sum_i \log(\lambda_i + \epsilon)$, where λ_i is the i-th eigenvalue of $\boldsymbol{G}$. Minimizing the heuristic effectively pushes λ_i towards zero thus achieving a solution with lower ranks.

In contrast to the trace norm of $\boldsymbol{G}$, the log*det* heuristic is concave. Therefore, minimizing it is not a convex optimization. Instead, we apply the first-order Taylor expansion around an existing solution $\boldsymbol{G}^t$,

$$\begin{aligned}\log\det(\boldsymbol{G} + \epsilon\boldsymbol{I}) \approx{}& \log\det(\boldsymbol{G}^t + \epsilon\boldsymbol{I}) \\ &+ \mathrm{Trace}[(\boldsymbol{G}^t + \epsilon\boldsymbol{I})^{-1}(\boldsymbol{G} - \boldsymbol{G}^t)]\end{aligned} \tag{13}$$

Note that the right-hand-side is a linear function in $\boldsymbol{G}$, thus convex. This approximation can be readily incorporated into the SDP formulation eq. (12) in searching for the low-rank solution of $\boldsymbol{G}$:

$$\begin{aligned}&\min \quad \mathrm{Trace}[\boldsymbol{G}\boldsymbol{L}] \\ &\qquad + \gamma\mathrm{Trace}[(\boldsymbol{G}^t + \epsilon\boldsymbol{I})^{-1}(\boldsymbol{G} - \boldsymbol{G}^t)] \\ &\text{s.t.} \quad 0 \leq G_{ij} \leq 1, G_{ii} = 1 \\ (\mathcal{LR} - \mathcal{SDP}) \quad &\boldsymbol{G}\mathbf{1} = \mathsf{N}/\mathsf{K}\,\mathbf{1} \\ &\boldsymbol{G} \succeq 0\end{aligned} \tag{14}$$

where γ is a tradeoff parameter between minimizing the conditional entropy (over the relaxed convex set) and preferences of low-rank solutions.

Solving the above optimization requires an iterative procedure that uses the minimizer as the new $\boldsymbol{G}^t$ and solves the problem again until convergence. To start the iteration at time $t = 0$, we use the SDP solution from eq. (12) without the log*det* heuristic as $\boldsymbol{G}^0$. In practice, the iterative procedure converges in a few iterations and often results in a solution whose rank is exactly the same as the number of clusters. The sketch of our algorithm is listed in Algorithm 1.

3.5 Partitions with uneven sizes

In deriving our SDP formulation of minimizing conditional entropies, we have assumed that every cluster has an equal number of data points. When this as-

Algorithm 1 Information Theoretic Clustering via Semidefinite Programming

Input: pairwise distance matrix $\boldsymbol{D}$,K,γ, $\epsilon \approx 10^{-4}$
Output: cluster membership

$\boldsymbol{L} \Leftarrow \log \boldsymbol{D}$ (elementwise logarithm)
$\boldsymbol{G}_0 \Leftarrow$ solve $\boldsymbol{G}$ according to eq. (12)
$t \Leftarrow 0$
while $\boldsymbol{G}^t$ not convernge **do**
  $t \Leftarrow t + 1$
  $\boldsymbol{L} = \log \boldsymbol{D} + \gamma(\boldsymbol{G}^t + \epsilon \boldsymbol{I})^{-1}$
  $\boldsymbol{G}^{t+1} \Leftarrow$ solve $\boldsymbol{G}$ according to eq. (12)
  $t \Leftarrow t + 1$
end while
Obtain top K eigenvalues and eigenvectors $\{(\lambda_k, \boldsymbol{v}_k)_{k=1}^{\mathsf{K}}\}$ from $\boldsymbol{G}^t$
For each $\boldsymbol{x_i}$, $\hat{\boldsymbol{a}}_i = [\sqrt{\lambda_1}\boldsymbol{v}_{1i}, ...\sqrt{\lambda_{\mathsf{K}}}\boldsymbol{v}_{\mathsf{K}i}]^\top$
cluster $\hat{\boldsymbol{a}}_i$ with K-means to obtain cluster membership

sumption is no longer valid, it is still possible to express the conditional entropy in terms of $\boldsymbol{G}$, and more specifically, Trace$[(\boldsymbol{G} - \boldsymbol{I})^{-1}\boldsymbol{L}]$, where $\boldsymbol{I}$ is the identity matrix. This is no longer a convex function of $\boldsymbol{G}$, therefore the relaxation procedure will not lead to a SDP.

When the number of points in each class is sufficiently large and there is no significant difference among the number of data points in each cluster, our assumption of equally sized partitions has minor effect and probably worths the price to pay in order to have a convex optimization. Note that during the final stage of our algorithm in computing the binary assignment matrix $\boldsymbol{A}$ (cf. section 3.3), the constraint of equal partitions is not enforced.

Additionally, it is possible to use our formulation to arrive at a good initial solution. The solution can then be subsequently refined by another procedure, for instance, a local search procedure that moves data points from one cluster to another cluster greedily. Such procedures will not enforce the constraint of equal partitions. Our preliminary results indicate that in some cases, this is a viable option, yielding better solutions than both the SDP solutions and solutions of local search without being initialized by the SDP solutions.

3.6 Optimization and computational complexity

Semidefinite programming is convex optimization and therefore can be solved efficiently in polynomial time and space complexity. The actual clock time, on the other hand, depends on several factors. First, interior-point based SDP solvers in general have a cubic complexity dependency on the number of constraints. In our formulations eq. (12) and eq. (14), the number of constraints are $O(\mathsf{N}^2)$ where N is the number of data points, mainly due to the constraints on the off-diagonal elements $G_{ij} \geq 0$. Therefore, the time complexity is at least $O(\mathsf{N}^6)$ and poses a significant challenge for applying the approach to large-scale problems. Given the constraints on our computational resources, we can solve problems with up to 200 data points with off-shelf SDP solvers.

Most solvers do not take advantage of the simple structures (such as elementwise nonnegativity) of our constraints. Therefore, it will be highly interesting and fruitful to develop specialized solvers for such problems. Particularly, as we will discuss in the following section, our problems are intimately connected to MAX K CUT problems on graphs, therefore, we expect such solvers to be of high values for those types of problems too.

4 RELATED WORK

Our work draws together different threads of research ideas. As a clustering algorithm, using mutual information (or equivalently conditional entropy) as a clustering criterion was first reported in (Faivishevsky and Goldberger, 2010).

The idea of using information theoretical measure for clustering can also be traced back to (Song et al., 2007), where they have employed a different estimation technique called Hilbert-Schmidt Independence Criterion (HSIC) to measure (in)dependency between random variables. Specifically, they map random variables to reproducing kernel Hilbert spaces and compute linear correlations between random functions. This estimation technique also circumvents the difficulty of estimating information theoretical quantities in high-dimensional spaces.

R. Gomes (2010) recently investigated the setting where the conditional distribution $P(Y|X)$ has a known parametric form. This enables a direct and analytic calculation of the mutual information between clustering labels and the data points, as opposed to the nonparametric estimator used in section 2.

Our work has also been inspired by semidefinite programming relaxation techniques for solving the MAX K CUT problems on weighted graphs. As shown in section 3, with mild assumptions, the information theoretical clustering can be cast as an integer programming and then subsequently relaxed into SDP similar to those used in graph cut problems (Goemans and Williamson, 1995). Our formulation does have a constraint that each cluster has an

equal number of data points. Such constraints, when enforced on graph cuts, are called balanced cut.

MAX K CUT problems are NP-hard. The most effective solution so far is based on semidefinite programming relaxation, which attains an approximation guarantee of $\left(1 - 1/\mathsf{K} + 2O(\log \mathsf{K})/\mathsf{K}^2\right)$ where K is the number of partitions. When $\mathsf{K} = 2$, the approximation guarantee is about 0.878. And it has also been shown that, improving the guarantee above $(0.878 + \epsilon)$ is NP-hard if the unique games conjecture (UGC) holds. In other words, it is highly likely, assuming UGC, the SDP relaxation is the best possible polynomial approximation algorithm.

The link between K-means clustering and MAX K CUT has long been noted. Various spectral and SDP relaxation techniques have been developed in similar vein as ours to solve K-means clustering as a combinatorial optimization (Zha et al., 2002; Xing and Jordan, 2003; Sugiyama et al., 2010; Bie and Cristianini, 2006)). In our experiments, we explore this strategy by replacing the objective function for information theoretical clustering from Trace[$\boldsymbol{GL}$] to that for K-means clustering Trace[$\boldsymbol{GD}$] where $\boldsymbol{D}$ is the pairwise distance matrix.

5 EXPERIMENTAL RESULTS

In the following, we report empirical studies of our proposed approaches for solving information theoretical clustering through semidefinite programming relaxation. We show the effectiveness of our approaches and contrast them to other clustering algorithms.

5.1 Setup

Datasets. We experimented on six datasets. Four datasets are from the UCI repository (Asuncion and Newman, 2007): *wine*, *iris*, *glass* and *vehicle*. The other two datasets are subsets of the USPS handwritten digits images. We have sampled a subset of 100 images, 20 each from the digits 1, 2, 3, 4 and 5 and another subset of 150 images, 30 each from the same set of digits. We refer the former as *usps-100* and the later as *usps-150*. Table. 1 summarizes basic characteristics of these datasets. Note that, the *wine* dataset is not balanced: each cluster has different numbers of data points: 59, 71, and 48. The *glass* data set is even more skewed, having 70, 76, 17, 13, 9, and 29 data points in its six classes.

Evaluation metric. We evaluate clustering results with the RAND index score (Rand, 1971), a standard nonparametric measure of clustering quality. RAND computes the agreements between two sets of different partitions, P_1 and P_2, of the same data set. Each partition is viewed as a collection of $N(N-1)/2$ pairwise decisions, where N is the size of the data set. For each pair of points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, they are either assigned to the same cluster or to different clusters. Let a_{ij} be the number of decisions where $\boldsymbol{x}_i$ is in the same cluster as $\boldsymbol{x}_j$ in P_1 and in P_2. Let b_{ij} be the number of decisions where the two instances are placed in different clusters in both partitions. Total agreement can then be calculated using

$$\mathrm{RAND}(P_1, P_2) = \frac{\sum_{ij}(a_{ij} + b_{ij})}{N(N-1)/2} \tag{15}$$

Table 1: Characteristics of the experimented datasets

Dataset	# of classes	Dimensions	Samples
wine	3	13	178
iris	3	4	150
glass	6	9	214
usps-100	5	256	100
usps-150	5	256	150
vehicle	4	18	847

For all the six datasets we have examined, we compute the RAND score between the outputs of clustering algorithms and the *ideal* clustering – induced by the labels of the data points, where data from the same class label are assigned to the same cluster.

Comparison We compare to the local search algorithm described in (Faivishevsky and Goldberger, 2010), the K-means clustering, and the SDP based K-means (described in section 4), referred as ITC-Local, K-means and K-means-SDP, respectively. We explore the option of solving the SDP formulation eq. (12) directly with SDPLR, an approximation algorithm developed in (Burer and Monteiro, 2003). Instead of identifying the optimal Gram matrix $\boldsymbol{G}$, the algorithm assumes a low-rank factorization of $\boldsymbol{G}$ and reformulates the convex SDP problem in nonconvex nonlinear optimization. While the algorithm is scalable to large problems, it does not solve eq. (12) exactly and in particular, not all the constraints in eq. (12) are satisfied. We term this algorithm, when applied to information theoretical clustering, as ITC-SDP-APPR, in contrast to ITC-SDP-Exact for our log*det* heuristic based algorithm (Algorithm 1).

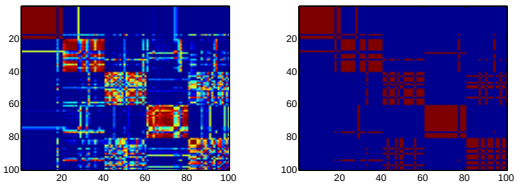
5.2 Effectiveness of low-rank refinement

We first examine the effectiveness of low-rank refinement described in section 3.4. This refinement algorithm iteratively updates the current estimation of $\boldsymbol{G}$, the Gram matrix computed from the cluster assignment matrix $\boldsymbol{A}$. The update is constructed such that low-rank solutions of $\boldsymbol{G}$ are preferred.

We initialized the iterative update with the solution to

the SDP eq. (12). After obtaining the refined solution from the heuristic augmented SDP eq. (14), we compare the sparsity patterns of the two solutions. We experimented on the data set *usps-100*, where 20 data points per class are *sequentially* ordered.

Fig. 1 illustrates the difference between the two matrices. The plots are images of the matrices' elements. Different colors show different values (ie, the inner products between $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ for the assignments of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$). Ideally, this matrix should contains two values 0 (assigning to different clusters) and 1 (assigning to the same cluster), for each pair of data points. The left plot shows the initial $\boldsymbol{G}$, ie, the SDP solution without the low-rank refinement. While there is a block-diagonal structure, the values of the matrix elements are clearly multi-modal, reflecting different degrees of cluster membership assignments. The right plot depicts a much cleaner block-diagonal structure, where most data points from the same (ground-truth) class labels are being assigned to the same clusters, reflected by the bimodal distributions of the inner products. Moreover, the matrix on the left has a rank of 14, while the matrix on the right has a rank of 5, precisely the number of classes in the data set. When the two matrices are used to derive binary assignment (cf. section 3.3), the low-rank matrix yields a RAND score of 0.921, significantly higher than the RAND score of 0.877 from the matrix on the left.



(a) $\boldsymbol{G}^0$ (b) $\boldsymbol{G}$ after low-rank refinement

Figure 1: Effect of low-rank refinement on the solution provided by SDP-based relaxation

5.3 Comparison to other clustering algorithms

We compare our proposed approach to other clustering algorithms in terms of RAND scores. To have errorbars on the RAND scores, for each of the six data sets, we ran 10 rounds of clustering. In each run, we randomly selected 95% of the data. And for all methods, we restart several times and choose the best result. Fig. 2 show the RAND scores of these methods. One standard deviation of the scores are plotted as error bars, deviating from the mean values.

RAND scores As explained previously, we have two

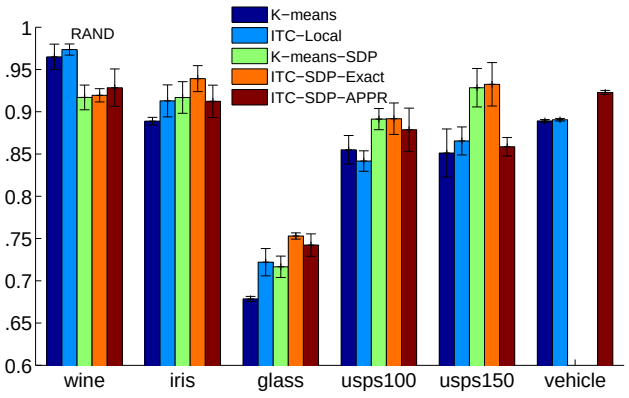


Figure 2: RAND scores of several clustering algorithms on the datasets listed in table 1. The higher the score, the better the clustering quality is. For the *vehicle* data set, only ITC-SDP-APPROX is reported. Other SDP-based methods did not return solutions within reasonable amount of times.

approaches of solving ITC with SDP relaxation, ITC-SDP-Exact and ITC-SDP-APPR. The former solves SDPs exactly with the low-rank heuristic (eq. (14) and the later solves eq. (12 approximately. Both approaches outperform other methods except on the *wine* dataset. One possible reason is that the dataset has an unbalanced number of data points in each cluster. However, on the dataset *glass* which is much more skewed, our approaches still perform the best.

Interestingly, the algorithm of SDP-relaxation based K-means (K-means-SDP) performs nearly as well as ITC-SDP-Exact and ITC-SDP-APPR, and in general performs better than the standard K-means algorithm, except on the *wine* dataset. This seems to suggest that, for at least 5 of the 6 data sets, the clustering criterion (be Euclidean distance based or conditional entropy based) has less significant effect than the optimization algorithm used to solve the clustering problem.

The information theoretical clustering algorithm (ITC-Local) originally proposed in (Faivishevsky and Goldberger, 2010)) did not perform as well as one would have expected, except in the *wine* dataset where it beats all other methods. This algorithm has significant computational advantages and in general, converge to a local optimum fast. Therefore, its value is likely to be more appreciated for large-scale clustering problems, which pose significant challenges for SDP based approaches.

Of the six datasets, the dataset *vehicle* is the largest. Only two optimization algorithms for ITC are able to complete: ITC-Local and ITC-SDP-APPR. While ITC-SDP-APPR returns only approximate solutions

to our SDP formulation of ITC (eq. (12), the algorithm still attains a higher RAND score than the ITC-Local does. This suggests strongly the virtue of exploiting SDP relaxation for ITC.

Conditional entropy. It is also interesting to compare the conditional entropies computed from various clustering results. Fig. 3 illustrate the differences among these methods, by displaying relatively how the conditional entropy has improved over the "ideal" clustering – treating class labels as ground-truth clusters memberships. To our surprise, on many data sets, most methods have similar conditional entropies despite significant differences in RAND scores. It seems that conditional entropies are only of limited correlation to RAND scores. For example, on the *glass* dataset, while the conditional entropies for all methods have been reduced from the "ideal" clustering, none of the methods attains very good RAND scores (cf. Fig. 2).

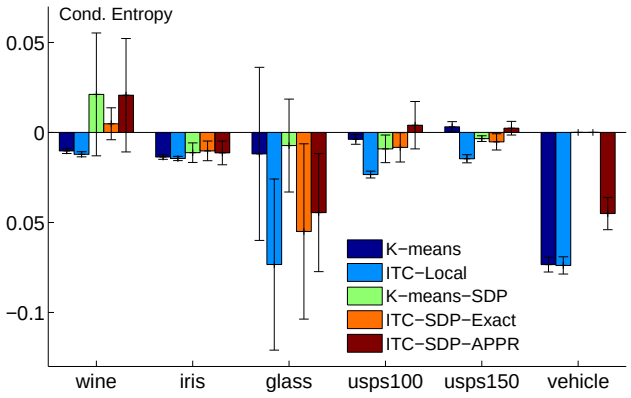


Figure 3: Relative improvement of conditional entropies over the conditional entropies computed from the "ideal clustering".

5.4 Robustness to numerical optimizations

Our low-rank heuristic based SDP eq. (14) depends on two parameters ϵ and γ. The parameter ϵ needs to be set very small and we set it around 10^{-4}. The parameter γ trades off between two components of the objective function: the conditional entropy and the log*det* heuristic. For example, a large γ will prefer a solution whose rank is less than the number of clusters. In practice, we have found that it is not difficult to choose a suitable γ. For example, on the *usps100* dataset, suitables γ span from 0.5 to 10.

With a properly chosen γ, eq. (14) is typically solved in about 10-15 iterations while each iteration solves a SDP problem. We have always initialized the iteration with the solution from eq. (12), ie, our SDP formulation without low-rank heuristic. That works in general well.

6 CONCLUSION

In this paper, we have investigated convex optimization techniques for information theoretical clustering, a recently proposed criterion in lieu of the standard K-means clustering. The clustering objective is to maximize the mutual information between data points and cluster memberships. We formulate the problem as an integer programming and apply semidefinite programming (SDP) relaxation. We also show how the SDP solution can be further improved by a low-rank heuristic. The refined solution reveals much more clearly the cluster structure in the data. On several datasets we have experimented, the proposed approach outperforms other methods in the standard evaluation metric of clustering quality.

There are several future directions to pursue. SDPs are still not practical for large-scale problems. This is especially true in our case where we aim to solve problems with a large number of constraints – quadratic in the number of data points. Exploiting the structures and the simplicity of the constraints are likely to be successful. This paper empirically studied the utility a SDP approximate solver and the preliminary results seem encouraging. Yet, another possibility is to reformulate the problem of minimizing conditional entropy as energy minimization where approximate graph cut algorithms can be applied Boykov et al. (2001)[1].

Through our experimental studies, we have also revealed an interesting phenomenon: despite being well-motivated, the mutual information being optimized has very limited correlations to the clustering quality. It is unclear why different methods, attaining similar mutual information, have significantly different clustering qualities. We are actively pursuing along this direction. One possibility is that the particular formulation through SDP relaxation has inductive bias for better clusterings and should not be viewed merely as an alternative optimization algorithm for information theoretic clustering[2]. Elucidating these is a subject of our future research.

References

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2001.

L. Faivishevsky and J. Goldberger. A nonparametric information theoretic clustering algorithm. In *Proceedings of*

[1]We thank reviewers for suggesting this promising direction.

[2]We thank reviewers for pointing these perspectives.

the 27th international conference on Machine learning, pages 351–358. ACM, Omnipress, 2010.

Q. Wang, S.R. Kulkarni, and S. Verdú. Divergence estimation for multidimensional densities via k-nearest-neighbor distances. *IEEE Transactions on Information Theory*, 55(5):2392–2405, 2009. ISSN 0018-9448.

L. Kozachenko and N. Leonenko. On statistical estimation of entropy of random vector. *Problems Infor. Transmiss*, 23(2):95–101, 1987.

Lev Faivishevsky and Jacob Goldberger. Ica based on a smooth estimation of the differential entropy. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 433–440, 2009.

V.V. Vazirani. *Approximation algorithms*. Springer Verlag, 2001. ISBN 3540653678.

D.S. Hochbaum. Approximation algorithms for NP-hard problems. *ACM SIGACT News*, 28(2):40–52, 1997. ISSN 0163-5700.

M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995. ISSN 0004-5411.

A. Frieze and M. Jerrum. Improved approximation algorithms for maxk-cut and max bisection. *Algorithmica*, 18(1):67–81, 1997. ISSN 0178-4617.

S.P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge Univ Pr, 2004. ISBN 0521833787.

B. Borchers and J.G. Young. Implementation of a primal–dual method for SDP on a shared memory parallel architecture. *Computational Optimization and Applications*, 37(3):355–369, 2007. ISSN 0926-6003.

L. Engebretsen, P. Indyk, and R. O'Donnell. Derandomized dimensionality reduction with applications. In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, page 712. Society for Industrial and Applied Mathematics, 2002. ISBN 089871513X.

M. Fazel, H. Hindi, and S.P. Boyd. Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices. In *American Control Conference, 2003. Proceedings of the 2003*, volume 3, pages 2156–2162. IEEE, 2003. ISBN 0780378962.

L. Song, A. Smola, A. Gretton, and K.M. Borgwardt. A dependence maximization view of clustering. In *Proceedings of the 24th international conference on Machine learning*, pages 815–822. ACM, 2007.

A. Krause R. Gomes. Discriminative clustering by regularized information maximization. In *Advances in Neural Information Processing Systems 24*, 2010.

H. Zha, X. He, C. Ding, M. Gu, and H. Simon. Spectral relaxation for k-means clustering. *Advances in Neural Information Processing Systems*, 2:1057–1064, 2002. ISSN 1049-5258.

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

M. Sugiyama, I. Takeuchi, T. Suzuki, T. Kanamori, H. Hachiya, and D. Okanohara. Conditional Density Estimation via Least-Squares Density Ratio Estimation. In *Proceedings 14th international conference on AI and statistics (AISTATS)*, 2010.

T.D. Bie and N. Cristianini. Fast SDP relaxations of graph cut clustering, transduction, and other combinatorial problems. *The Journal of Machine Learning Research*, 7:1409–1436, 2006. ISSN 1532-4435.

A. Asuncion and DJ Newman. UCI Machine Learning Repository. Irvine, CA: University of California. *School of Information and Computer Science*, 12, 2007.

W.M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971. ISSN 0162-1459.

Samuel Burer and Renato D. C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Math. Program.*, 95(2):329–357, 2003.

Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.