10-715 Project Report: Information Theoretic Clustering using Kernel Density Estimation

Shashank Singh
Department of Statistics
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213
sss1@andrew.cmu.edu

Bryan Hooi
Department of Statistics
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213
bhooi@andrew.cmu.edu

1 Introduction

In recent years, information-theoretic clustering algorithms have been proposed which assign data points to clusters so as to maximize the mutual information between cluster labels and data [1, 2]. Using mutual information for clustering has several attractive properties: it is flexible enough to fit complex patterns in the data, and allows for a principled approach to clustering without assuming an explicit probabilistic generative model for the data.

Recently, [3] showed examples in which maximizing mutual information leads to poor performance when the clusters are unbalanced, even in very simple cases. This occurs because algorithms that maximize mutual information are biased toward splitting the data into equal-sized clusters, and this causes the algorithms to perform poorly in the large data limit. They propose to instead minimize the conditional entropy of the cluster labels given the data, where conditional entropy is estimated using a nearest neighbor-based approach. They show that minimizing this objective instead of maximizing mutual information tends to select clusters with large gaps separating them rather than simply selecting approximately equal-sized clusters.

In terms of computation, [2] formulate the mutual information maximization problem as a semidefinite program, and additionally use a low rank inement heuristic to obtain low rank solutions. [3] apply a similar approach to minimizing conditional entropy, involving a relaxation of the discrete cluster variables resulting in a semidefinite program, which is then converted to a candidate solution by rounding with respect to random projections, taking a similar approach to [4].

2 Proposal

In our project, we aim to further develop the clustering approach of [3] by estimating conditional entropy using a kernel density based approach rather than a nearest neighbor-based approach. This has several motivations:

1. Reducing variance: the k-nearest neighbor approach for estimating quantities (for fixed k) generally has nonzero variance in the large data limit, and hence performs poorly compared to consistent methods.
2. The theoretical properties of kernel density estimation are fairly well-studied, and we hope to use this to answer questions about the resulting information theoretic clustering algorithm, such as its performance in low and high dimensional settings.

The kernel-based estimator for conditional entropy we plan to use is as follows. Letting $K(\cdot)$ be the kernel:

$$\hat{H}(Y|X) = \frac{1}{n}\sum_{i=1}^{n} \log \frac{\hat{p}(x_i, y_i)}{\hat{p}(x_i)} \tag{1}$$

$$= \frac{1}{n}\sum_{i=1}^{n} \log \left(\frac{\frac{1}{hn}\sum_{j=1}^{n} 1\{y_j = y_i\} K(\frac{x_i - x_j}{h})}{\frac{1}{hn}\sum_{j=1}^{n} K(\frac{x_i - x_j}{h})} \right) \tag{2}$$

We also aim to investigate efficient algorithms for computing information-theoretic clustering. We aim to first adapt a similar method as the semidefinite program of [3] to our situation (where our objective involves kernel density estimation rather than a nearest neighbor-based estimate of conditional entropy), and then investigate whether the objective can be computed more efficiently using alternate methods.

Finally, we will test our algorithm on various examples such as those in [3], as well as low and high dimensional examples, in order to verify that it performs well in these settings. We will then evaluate the algorithm on datasets such as the UCI Machine Learning database *glass*, *iris* and *wine*[5] datasets, comparing our algorithm with other clustering algorithms in terms of RAND index score [6]. The other algorithms would include the K-means algorithm, mutual information-based algorithms such as [2], as well as [3], which also minimizes conditional entropy but uses a nearest neigbors-based approach.

3 Plan of Activities

Before the midterm report, we aim to:

1. Implement the kernel density based estimator for conditional entropy
2. Adapt the semi-definite program based approach for minimizing conditional entropy in [3] to our setting
3. Preliminary results for investigating some of the theoretical properties of the information theoretic clustering algorithm
4. Finalize datasets and evaluation metrics and algorithms to compare against, and start obtaining empirical results

4 Conditional Entropy and the K-Means Algorithm

In this section, we draw a connection between minimizing conditional entropy and the K-means algorithm. To do this, we construct an upper bound for $\hat{H}(X|Y)$. We then show that when a Gaussian kernel is used, this upper bound becomes the usual K-means objective, and hence the K-means algorithm can be seen as a way to minimize an upper bound on the estimated conditional entropy $\hat{H}(X|Y)$. This will allow us to obtain an upper bound on $\hat{H}(Y|X)$ as well.

Let $C_k = \{i : Y_i = k\}$, the indices of points assigned to cluster k.

Theorem 1. *When using a Gaussian kernel function $K(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}$, the estimated conditional entropy $\hat{H}(X|Y)$ satisfies:*

$$\hat{H}(X|Y) \leq \log(h) + \frac{1}{2}\log(2\pi) + \frac{1}{2h^2 n}\sum_{k=1}^{K}\sum_{i \in C_k}(x_i - \mu_k)^2$$

Consequently, minimizing the K-means objective $\sum_{k=1}^{K}\sum_{i \in C_k}(x_i - \mu_k)^2$ is equivalent to minimizing an upper bound for $\hat{H}(X|Y)$.

Proof. We estimate $\hat{H}(X|Y)$ as:

$$\hat{H}(X|Y) = -\frac{1}{n}\sum_{i=1}^{n}\log\hat{p}(x_i|y_i)$$

Estimating $\hat{p}(x_i|y_i)$ using kernel density estimation restricted to points assigned to the same cluster as y_i:

$$\begin{aligned}\hat{H}(X|Y) &= -\frac{1}{n}\sum_{i=1}^{n}\log(\frac{1}{hn_{y_i}}\sum_{j=1}^{n}1\{y_i=y_j\}K(\frac{x_i-x_j}{h})) \\ &= \log(h) - \frac{1}{n}\sum_{i=1}^{n}\log(\frac{1}{n_{y_i}}\sum_{j=1}^{n}1\{y_i=y_j\}K(\frac{x_i-x_j}{h}))\end{aligned}$$

The argument to the $\log$ is an average over terms, of which n_{y_i} terms are nonzero. Hence by Jensen's inequality on the convex function $f(x) = -\log(x)$ we interchange the log and the averaging operation:

$$\hat{H}(X|Y) \leq \log(h) - \frac{1}{n}\sum_{i=1}^{n}\frac{1}{n_{y_i}}\sum_{j=1}^{n}1\{y_i=y_j\}\log(K(\frac{x_i-x_j}{h}))$$

Now using the Gaussian kernel $K(x) = \frac{1}{\sqrt{2\pi}}e^{\frac{-x^2}{2}}$:

$$\begin{aligned}\hat{H}(X|Y) &\leq \log(h) - \frac{1}{n}\sum_{i=1}^{n}\frac{1}{n_{y_i}}\sum_{j=1}^{n}1\{y_i=y_j\}(-\frac{1}{2}\log(2\pi) - \frac{1}{2}(\frac{x_i-x_j}{h})^2) \\ &= \log(h) + \frac{1}{2}\log(2\pi) + \frac{1}{n}\sum_{i=1}^{n}\frac{1}{n_{y_i}}\sum_{j=1}^{n}1\{y_i=y_j\}(\frac{1}{2}(\frac{x_i-x_j}{h})^2) \\ &= \log(h) + \frac{1}{2}\log(2\pi) + \frac{1}{2h^2n}\sum_{i=1}^{n}\frac{1}{n_{y_i}}\sum_{j=1}^{n}1\{y_i=y_j\}(x_i-x_j)^2\end{aligned}$$

The double summation sums $(x_i - x_j)^2$ for exactly the ordered pairs (i, j) where i and j are assigned to the same cluster. Rearranging to sum over clusters:

$$\hat{H}(X|Y) \leq \log(h) + \frac{1}{2}\log(2\pi) + \frac{1}{2h^2n}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i,j\in C_k}(x_i-x_j)^2 \tag{3}$$

To compare this to the K-means objective, let μ_k be the centroid of cluster k, then:

$$\begin{aligned}\sum_{i,j\in C_k}(x_i-x_j)^2 &= \sum_{i\in C_k}\sum_{j\in C_k}(x_i-\mu_k+\mu_k-x_j)^2 \\ &= \sum_{i\in C_k}\sum_{j\in C_k}((x_i-\mu_k)^2+(x_j-\mu_k)^2)+0\end{aligned}$$

in the above the cross terms sum to 0 when summing over j for any fixed i. Thus this becomes:

$$\sum_{i,j\in C_k}(x_i-x_j)^2 = 2n_k\sum_{i\in C_k}(x_i-\mu_k)^2$$

substituting this into (3):

$$\hat{H}(X|Y) \leq \log(h) + \frac{1}{2}\log(2\pi) + \frac{1}{h^2n}\sum_{k=1}^{K}\sum_{i\in C_k}(x_i-\mu_k)^2$$

We thus see that minimizing the K-means objective $\sum_{k=1}^{K}\sum_{i\in C_k}(x_i-\mu_k)^2$ is equivalent to minimizing an upper bound on the conditional entropy estimator $\hat{H}(X|Y)$. □

An intuitive interpretation of this result comes by considering the equality conditions of the Jensen's inequality step: since $f(x) = -\log(x)$ is strictly convex, equality holds in the Jensen's inequality step iff $(x_i - x_j)^2$ is equal for all i, j in the same cluster. In other words, when the pairwise distances between points in the same cluster are around equal, minimizing the K-means objective and $\hat{H}(X|Y)$ become roughly equivalent (however, this should be seen as an approximation since the cluster assignments are not fixed). As the within-cluster pairwise distances become more uneven, Jensen's inequality becomes stricter and the K-means objective grows faster, suggesting that the K-means algorithm differs from conditional entropy minimization in that K-means prefers tighter clusters in which the pairwise distances are more evenly distributed.

[2] shows that when estimating $\hat{H}(X|Y)$ using a k-nearest neighbors based approach instead of the kernel density based approach used here, $\hat{H}(X|Y)$ is given by (up to a constant):

$$\sum_{k=1}^{K} \frac{d}{n(|C_k| - 1)} \sum_{i,j \in C_k} \log \|x_i - x_j\|_2^2$$

which is similar to our expression; however, it has additional logarithms on the squared distances, and thus grows more slowly with distance.

We now consider minimizing the conditional entropy $\hat{H}(Y|X)$:

Corollary 1. *The estimated conditional entropy $\hat{H}(Y|X)$ satisfies:*

$$\begin{aligned}\hat{H}(Y|X) &\leq \hat{H}(Y) + \log(h) + \frac{1}{2}\log(2\pi) + \frac{1}{h^2 n}\sum_{k=1}^{K}\sum_{i \in C_k}(x_i - \mu_k)^2 \\ &\leq -\sum_{k=1}^{K}\frac{|C_k|}{n}\log\frac{|C_k|}{n} + \log(h) + \frac{1}{2}\log(2\pi) + \frac{1}{h^2 n}\sum_{k=1}^{K}\sum_{i \in C_k}(x_i - \mu_k)^2\end{aligned}$$

Proof.

$$\hat{H}(Y|X) = \hat{H}(Y) + \hat{H}(X|Y) - \hat{H}(X)$$

$\hat{H}(X)$ depends only on the data X and can be ignored. The result then follows from theorem 1 in addition to writing $\hat{H}(Y) = -\sum_{k=1}^{K}\frac{|C_k|}{n}\log\frac{|C_k|}{n}$. □

Thus, minimizing $\hat{H}(Y|X)$ is similar to minimizing $\hat{H}(X|Y)$, except that the additional $\hat{H}(Y)$ means we prefer clusterings with lower $\hat{H}(Y)$. [3] notes that minimizing $\hat{H}(X|Y)$, or equivalently maximizing mutual information $I(X;Y) = H(X) - H(X|Y)$, leads to solutions with excessively similar number of points in each cluster and hence poor performance, and that minimizing $\hat{H}(Y|X)$ provides better empirical performance.

5 Conditional Entropy Minimization via Gradient Descent

6 More Theory

Not affected by the number of points in each cluster, but dependent on the variance of the two clusters; that is, it is biased towards clusterings of equal entropy.

An important special case is when

Fix a number of clusters $K \in \mathbb{N}$, let $p_1, \ldots, p_K$ be distributions on a sample space $\mathcal{X}$ and let $\pi = (\pi_1, \ldots, \pi_K)$ be a distribution on $[K]$. Consider random variables $x_1, \ldots, x_n$ in $\mathcal{X}$ drawn independently from the following process:

1. Draw $y_i \sim \pi$

2. Draw $x_i \sim p_{y_i}$

Then, the problem of clustering is to estimate $y_1, \ldots, y_n$ given $x_1, \ldots, x_n$.

Without any further assumptions, this is, of course, impossible; for any true labels $y_1, \ldots, y_n$, there are distributions $p_1, \ldots, p_K$ under which the observations $x_1, \ldots, x_n$ are highly likely.

If we were given the true cluster distributions $p_1, \ldots, p_K$, a reasonable and trivial maximum likelihood approach would be to assign $y_i = \arg\max p_y(x_i)$, but estimating $p_1, \ldots, p_k$ precisely is a difficult problem

Because clustering is, in general, an ill-defined problem, it is crucial, for any clustering algorithm, to understand the nature of the clusters extracted by that algorithm. Suppose the data points are drawn from K underlying distributions, $p_1, \ldots, p_k$, and let n_i denote the number of points drawn from p_i ($i \in [K]$). Then, the clustering problem can be well defined as finding a partition $\mathcal{P} = \{S_1, \ldots, S_K\}$ of $\{x_i : i \in [n]\}$ such that $\mathcal{P} = \{\{x_i : x_i \sim p_k\} : k \in [K]\}$, or at least minimizing some loss function over the space of partitions of $\{x_i : i \in [n]\}$.

Clustering via Mutual Information Maximization will perform well when $n_1 = \cdots n_k$. Clustering via Conditional Entropy Minimization, on the other hand, will perform well when $H(p_1) = \cdots = H(p_k)$.

$$I(X;Y) = H(X) - H(X|Y) = H(p) - \sum_{y=1}^{k} \pi_y H(p_y).$$

$$H(Y|X) =$$

7 The Relationship Between MIMax and CHMin

Naïvely minimizing $H(Y|X)$ will place all data points in a single cluster (so that $H(Y|X) \leq H(Y) = 0$). Since

$$I(X;Y) = H(Y) - H(Y|X) = \log(K) - (D_{KL}(Y,U) + H(Y|X)),$$

where U is distributed uniformly over $[K]$,

$$\arg\max_Y I(X;Y) = \arg\min_Y H(Y|X) + D_{KL}(Y,U).$$

Hence, MIMax can be thought of as regularizing CHMin by penalizing label distributions far from uniform; said another way, relative to CHMin, MIMax generates clusters of equal (sample) size.

8 Empirical Results

We implemented the gradient descent algorithm described above and tested it on gradient

The number of clusters was ASSUMED TO BE KNOWN BEFOREHAND OR CHOSEN TO MINIMIZE THE NUMBER OF ITERATIONS.

8.1 Algorithms

8.2 Datasets

We tested clustering algorithms on 6 datasets, of which 3 were synthetic and 3 were real datasets taken from the UCI repository.

8.2.1 Synthetic Datasets

Three Gaussians: 100 samples from each of spherical 3 Gaussians in $\mathbb{R}^3$, with covariance $0.25I$ and means $(0,0,1)$, $(0,1,0)$, and $(1,0,0)$. Here, we expect all algorithms to perform very well, given the reasonably distinct, spherical clusters.

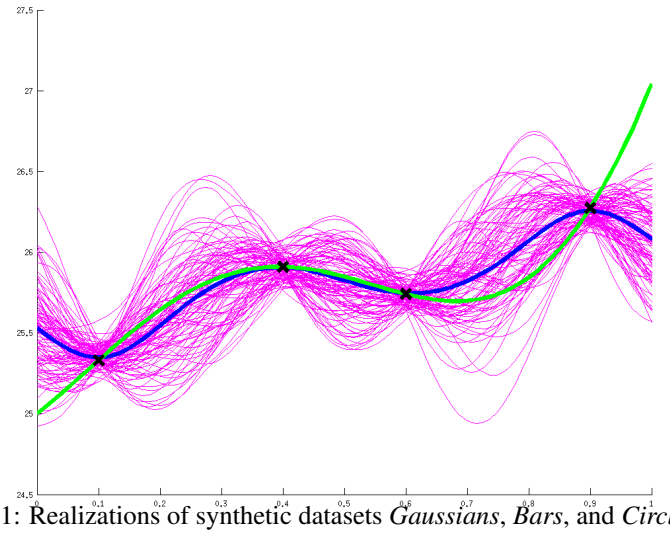

Figure 1: Realizations of synthetic datasets *Gaussians*, *Bars*, and *Circles*.

Dataset (# clusters)	CHMin	K-means++	HC (single)	HC (complete)	HC (average)
Gaussians (3)	0.991 ± 0.008	**0.998**	0.767	0.984	0.994
Bars (2)	$\mathbf{0.723} \pm 0.013$	0.520	0.502	0.524	0.524
Circles (2)	$\mathbf{0.894} \pm 0.019$	0.671	0.501	0.677	0.605
Iris (3)	$\mathbf{0.929} \pm 0.031$	0.893	0.680	0.840	0.906
Wine (3)	0.675 ± 0.0283	**0.702**	0.427	0.674	0.612
Wine5 (3)	$\mathbf{0.700} \pm 0.044$	0.494	0.387	0.500	0.500

Table 1: Mean ($\pm$ standard deviation) of accuracy, for each algorithm on each *dataset* (# of clusters). Hierarchical clustering algorithms are denoted HC (type), where type is the linkage procedure used. Means and standard deviations were computed from 100 trials, with synthetic datasets resampled in each trial. Boldface numbers indicate the best performing algorithm for each dataset.

Bars: 400 noisy samples (x_i, y_i) from each of two long, vertical lines, where each $y_i \sim \text{Unif}(0, 10)$, $x_1, \ldots, x_{200} \sim \mathcal{N}(0, 0.3)$, and $x_{201}, \ldots, x_{400} \sim \mathcal{N}(1, 0.3)$. We expect K-means to perform poorly here because the variance in y is much greater than in x, while only x depends on the cluster.

Circles: 200 noisy samples $(x_i, y_i) = (r_i \cos\theta_i, r_i \sin\theta_i)$ from each of two concentric circles, where $\theta_i \sim \text{Unif}(0, 2\pi)$, $r_1, \ldots, r_{200} \sim \mathcal{N}(1, 0.3)$, and $r_{201}, \ldots, r_{400} \sim \mathcal{N}(2, 0.3)$.

8.2.2 Real Datasets

Iris:

Wine: Due to the large number of features relative to the sample size, we expect the kernel density estimation step may cause CHMin may perform poorly.

Wine5: A subset of *Wine* using the (arbitrarily chosen) first 5 features and all 178 data points. This subset is used to study how the performance of CHMin scales with the dimension of the data.

8.3 Results

Of the five algorithms tested, CHMin performed the best on 4 of the 6 datasets. For the remaining two datasets (*Gaussians* and *Wine*), K-means++ performed best. Since *Gaussians* is precisely the sort of scenario for which K-means is designed, this is unsurprising. We suspect the relatively poor behavior of CHMin on *Wine* is due to the poor behavior of kernel density estimation in higher dimensional spaces (the curse of dimensionality). This is supported by the fact that the performance of CHMin improved when the number of features in *Wine* was reduced to 5, whereas the perfor-

mance of all other algorithms fell sharply. For both *Gaussians* and *Wine*, CHMin still performed competitively (within 3 percentage points of K-means).

Bandwidth and Step Size Selection

Runtime

As mentioned above, the naive runtime of CHMin is quadratic in the number of samples. However, using the k-d tree data structure to maintain the kernel density estimate, in addition to using either a kernel with bounded support or an appropriate approximation of a kernel with unbounded support, the runtime can be reduced to $O(n \log n)$. HC algorithms take time quadratic in the sample size, while K-means and its variants take only linear time.

9 Conclusions

Works well as long as the dimension is not too large.

References

[1] Lev Faivishevsky and Jacob Goldberger. Nonparametric information theoretic clustering algorithm. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 351–358, 2010.

[2] Meihong Wang and Fei Sha. Information theoretical clustering via semidefinite programming. In *International Conference on Artificial Intelligence and Statistics*, pages 761–769, 2011.

[3] Greg Ver Steeg, Aram Galstyan, Fei Sha, and Simon DeDeo. Demystifying information-theoretic clustering. *arXiv preprint arXiv:1310.4210*, 2013.

[4] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[5] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[6] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.