Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Pairwise Exemplar Clustering

Yingzhen Yang[1] **Xinqi Chu**[1] **Feng Liang**[2] **Thomas S. Huang**[1]
Department of Electrical and Computer Engineering[1], Department of Statistics[2]
University of Illinois at Urbana-Champaign

Abstract

Exemplar-based clustering methods have been extensively shown to be effective in many clustering problems. They adaptively determine the number of clusters and hold the appealing advantage of not requiring the estimation of latent parameters, which is otherwise difficult in case of complicated parametric model and high dimensionality of the data. However, modeling arbitrary underlying distribution of the data is still difficult for existing exemplar-based clustering methods. We present Pairwise Exemplar Clustering (PEC) to alleviate this problem by modeling the underlying cluster distributions more accurately with non-parametric kernel density estimation. Interpreting the clusters as classes from a supervised learning perspective, we search for an optimal partition of the data that balances two quantities: 1 the misclassification rate of the data partition for separating the clusters; 2 the sum of within-cluster dissimilarities for controlling the cluster size. The broadly used kernel form of cut turns out to be a special case of our formulation. Moreover, we optimize the corresponding objective function by a new efficient algorithm for message computation in a pairwise MRF. Experimental results on synthetic and real data demonstrate the effectiveness of our method.

Introduction

Clustering is an important data analysis method which partitions data space into a set of self-similar clusters. Representative clustering methods include K-means which finds a local minima of sum of with-cluster dissimilarities, spectral clustering (Ng, Jordan, and Weiss 2001) which identifies clusters of more complex shapes lying on some low dimensional manifolds, and statistical modeling of the data by a mixture of parametric distribution (Fraley and Raftery 2002). Among them, exemplar-based clustering methods such as Affinity Propagation (Frey and Dueck 2007) are appealing since they do not need to estimate latent parameters while combining data partition and model selection for the number of clusters in the same optimization scheme.

However, it is difficult for exemplar-based methods to characterize arbitrary underlying distributions of the data. For example, pairwise similarity measures are not enough for Affinity Propagation to recover the structure of the data with a combinatorial optimization algorithm. Recent works on exemplar-based clustering try to alleviate this problem via various statistical modeling techniques. (Tarlow, Zemel, and Frey 2008) imposed Dirichlet process priors on the distribution of cluster sizes, but such priors on the distribution of cluster sizes are not always effective in revealing the underlying distribution of the data itself. Both (Lashkari and Golland 2007) and its accelerated version (Takahashi 2011) fit a mixture of exponential family distributions to the data and restrict the mean of the mixture components to the set of data points, taking advantage of the exemplar finding. Although they formulated a convex optimization problem by taking only the weight of mixture components as variables, the parametric assumption about the data distribution limits the potential of their methods. On the other hand, non-parametric clustering methods (Li, Ray, and Lindsay 2007; Comaniciu and Meer 2002; Hinneburg and Gabriel 2007) exhibit the power of kernel density estimators in modeling the data distribution and clustering. They are similar in searching for the local modes of the kernel density and grouping the data points that climb to nearby modes together, where a heuristic threshold for merging the modes is always needed (Xu and II 2005). Nevertheless, these kernel based methods lack a unified optimization scheme for both grouping the data and choosing the number of clusters, so that they cannot avoid the heuristic mode merging process.

Combining the advantages of kernel methods and exemplar-based clustering scheme, we propose a new clustering method, Pairwise Exemplar Clustering (PEC). The assumption is that the given dissimilarities between data points is their norm distances in Euclidean space. Compared to traditional exemplar-based clustering methods, PEC approximates the underlying cluster distributions more accurately by kernel density estimation. Inspired by the connection between supervised learning and unsupervised clustering (Xu et al. 2004; Gomes, Krause, and Perona 2010), we derive a misclassification rate of any hypothetical data partition with respect to nearest neighbor classifier. This misclassification rate is a new measure to evaluate the separability of the corresponding clusters, and the widely used kernel form of cut (Wu and Leahy 1993) becomes a special case of our formulation under the new measure. We convert the optimization of the new measure to a MAP problem in a pairwise MRF, and design a new algorithm for efficient message computa-

tion to greatly speedup the inference process in the pairwise MRF.

The rest part of this paper is organized as follows. We first introduce the misclassification rate of a data partition, then build the objective function for clustering followed by the optimization algorithm. After that, we demonstrate and analyze the experimental results, and finally conclude the paper.

The Proposed Pairwise Exemplar Clustering

Before formulating our clustering method, we introduce the notations in the following formulation. Suppose the data set $X = (x_1, x_2, ...x_N)$ belongs to the D-dimensional Euclidean space R^D. Given their pairwise dissimilarities $[d_{ij}]_{i,j=1:N}$, where d_{ij} is the norm distance between x_i, x_j, i.e. $d_{ij} = \|x_i - x_j\|$, PEC associates each data point x_i with a cluster indicator c_i ($i \in \{1, 2, ...N\}, c_i \in \{1, 2, ...N\}$), indicating that x_i takes x_{c_i} as the cluster exemplar. We define $c = \{c_i\}_{i=1}^N$. Furthermore, our clustering algorithm partitions X into Q disjoint clusters $C = \{C_i\}_{i=1}^Q$, and we can get the partition C from cluster indicators c since all the data points with the same cluster indicator form a cluster.

Misclassification Rate of Data Partition

To avoid the restrictions posed by parametric distributions, we model the data by kernel density estimation. Given N data points $X = (x_1, x_2, ...x_N)$, and suppose they are i.i.d. samples drawn from some distribution with an unknown density function f, the variable bandwidth kernel density estimator at any point x is

$$\widehat{f}(x) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{h_i^D}K\left(\frac{x - x_i}{h_i}\right) \tag{1}$$

where we use the radially symmetric Gaussian kernel $K(x) \propto \exp\left(-\|x\|^2 \Big/ 2\right)$, and

$$h_i = h_0\left(\hat{\lambda} \Big/ \hat{f}_0(x_i)\right)^{\frac{1}{2}} \tag{2}$$

is the variable bandwidth at x_i by the sample point estimator (Abramson 1982) where h_0 is a fixed bandwidth. Suggested by (Silverman 1986), $\hat{\lambda}$ is the geometric mean of $\left\{\hat{f}_0(x_i)\right\}_{i=1}^N$ and $\hat{f}_0$ is chosen as the fixed bandwidth kernel density estimator with h_0. This setting for variable bandwidth is known to reduce the bias while remaining the variance of the MSE of the kernel density estimator (Abramson 1982). We prefer variable bandwidth density estimator due to its ability to model data with different scales. Thanks to the radially symmetric kernel, we can compute the variable bandwidth just by the given pairwise dissimilarities between data points.

Similar to (Li, Ray, and Lindsay 2007), the density estimate for cluster $C_j, j \in \{1, 2, ...Q\}$ is

$$\widehat{f_j}(x) = \frac{1}{|C_j|}\sum_{i=1}^{N}\frac{1}{h_i^D}K\left(\frac{x - x_i}{h_i}\right)\mathcal{I}_{C_j}(x_i) \tag{3}$$

where $\mathcal{I}$ is an indicator function. (3) is capable of describing much more complex cluster distributions than parametric statistical modeling (Terrell and Scott 1992). Since the variable bandwidth $\{h_i\}_{i=1}^N$ can be estimated before clustering, $\widehat{f_j}$ is entirely determined by C_j.

If viewing clusters as classes, there is a natural connection between clustering and multi-class classification from a supervised learning perspective (Gomes, Krause, and Perona 2010). For any hypothetical data partition C, we have a corresponding classification model $M = \left(\{f_j, \pi_j, C_j\}_{j=1}^Q, F\right)$ such that f_j is the density estimator for class C_j and $f_j = \widehat{f_j}$, π_j is the weight of C_j, and F is a classifier trained using the Q classes $\{C_j\}_{j=1}^Q$. We restrict F to be nearest neighbor classifier since it does not make any assumption about the distribution of the training data. Note that the our classification model is built from hypothetical data partition in an unsupervised manner rather than the training data with ground truth labels. Moreover, in many practical problems data points are generated from multiple unobserved classes, so it is particulary important for a clustering method to identify these underlying classes and assign the data points to the corresponding unobserved classes they come from. This problem is reduced to finding an optimal classification model that well separate the classes in our setting, and we prefer the data partition which minimizes the misclassification rate defined below (Bishop 2006):

Definition 1. *The misclassification rate of a classification model M is*

$$\widetilde{P}(M) = \sum_{j=1}^{Q}\int_{\bigcup_{i\neq j}\mathcal{R}_i} p(x, C_j)\, dx \tag{4}$$

where $p(x, C_j)$ is the joint distribution of the data x and class C_j, and $\mathcal{R}_i$ is the decision region of class C_i determined by the classifier F.

The misclassification rate of a data partition C is that of the classification model M corresponding to C. Clearly (4) is dependent on the decision regions determined by the classifier F. Finding the exact decision regions for all possible classifiers is prohibitively time consuming, so we aim to formulate the decision regions compatible with all classifiers instead. A conservative choice of $\mathcal{R}_i$ is $\mathcal{R}_i = C_i$. However, with this choice (4) would yield 0 for any classification model M. Inspired by the behavior of nearest neighbor classifier (Mclachlan 2004) and in order to compute the misclassification rate more accurately, we extend $\mathcal{R}_i$ from C_i to the δ-cover of C_i, which results in a new decision region $\mathcal{R}_i^\delta$ defined as

$$\mathcal{R}_i^\delta = \bigcup_{x_m \in C_i} B(x_m, \delta_m) \tag{5}$$

$B(x_m, \delta_m)$ is a D-dimensional ball with radius δ_m ($\delta_m > 0$) centered at x_m , and $\mathcal{R}_i$ is infinitely close to C_i when $\delta_m \to 0$ for all $x_m \in C_i$. The idea behind the δ-cover is that we can assign the unobserved data within the ball $B(x_m, \delta_m)$ to the same class as x_m in case of nearest

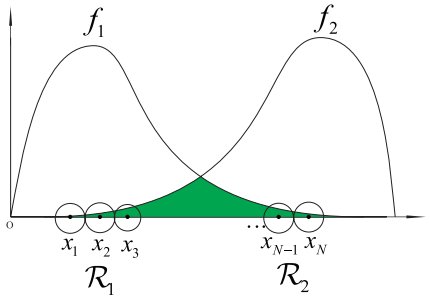

Figure 1: Two-class classification model with δ-cover

neighbor classifier, when the radius of the ball, δ_m, is small enough. Since any two data points could be assigned to different classes, we are safe to choose δ_m as large as half of the distance between x_m and its nearest neighbor, that is,

$$0 < \delta_m \leq \frac{d_{mm^*}}{2} \tag{6}$$

and m^* is the nearest neighbor of m.

Combining all the decision regions where (6) holds everywhere, we obtain a δ-cover of X, i.e. $B_\delta = \{B(x_m, \delta_m)\}_{m=1}^N$. We refer to the δ-cover of X such that $\delta_m = d_{mm^*}/2$ for any data point x_m as nearest neighbor δ-cover. We define the δ-misclassification rate of a classification model M as the misclassification rate on the δ-cover of X:

$$\widetilde{P}_\delta(M) \triangleq \sum_{j=1}^{Q} \int_{\bigcup_{i \neq j} \mathcal{R}_i^\delta} p(x, C_j)\, dx \tag{7}$$

Figure 1 shows the two-class classification model illustrating the δ-cover of X. We aim to minimize the δ-misclassification rate (7) so as to well separate different classes, the very objective of clustering. Although (7) is not a closed-form due to the integral, its upper bound can be obtained from Theorem 1.

Theorem 1. *Given data points $X = (x_1, x_2, ...x_N)$ and the hypothetical partition C on X, let M be the corresponding classification model, then for any δ-cover of X,*

$$\widetilde{P}_\delta(M) \leq \frac{c_0}{N} \left(\frac{\delta_{\max}}{h_{\min}} \right)^D \psi_\delta(c) \tag{8}$$

where

$$\psi_\delta(c) \triangleq \sum_{m=1}^{N} \sum_{l=1}^{N} \left(K\left(\frac{x_m - x_l}{h_l} \right) + \frac{G_{ml}\delta_m}{h_l} \right) \theta_{lm} \tag{9}$$

$\theta_{lm} = \mathcal{I}_{\{c_l \neq c_m\}}$, G_{ml} is an upper bound for $\|\nabla K((x - x_l)/h_l)\|$ constrained within the ball $B(x_m, \delta_m)$ [1], c_0 is a constant, $h_{\min} = \min\{h_i\}_{i=1}^N$ and $\delta_{\max} = \max\{\delta_i\}_{i=1}^N$.

[1]Note that all continuous kernels have bounded gradient within a ball, and $G_{ml} \leq e^{-0.5}$ for the radially symmetric Gaussian kernel

The proof is shown in the appendix. We call ψ_δ the pairwise kernel density (PKD) term. According to (8), $\widetilde{P}_\delta(M)$ is bounded by ψ_δ up to a constant scale for any fixed δ-cover of X, so we can approximately minimize $\widetilde{P}_\delta(M)$ by minimizing its upper bound ψ_δ, which is more tractable.

Relationship to cut

Furthermore, it is interesting to observe the connection between $\psi_\delta(c)$ and the widely used kernel form of cut for clustering and segmentation (Wu and Leahy 1993). The cut defined on graph $\widetilde{G}(V, E)$ is:

$$cut(A, B) = \sum_{i \in A, j \in B} W(i, j) \tag{10}$$

and the literature broadly adopts the Gaussian kernel function $W(i,j) = \exp\left(-\|x_i - x_j\|^2 \Big/ 2h^2\right)$ (x_i and x_j are the feature vectors of node i and j), to represent the similarity between i, j. The relationship between ψ_δ and cut is described below:

Remark 1. *If $h_i = h$ for $i \in \{1, 2, ...N\}$ and $W(i,j) = K((x_i - x_j)/h)$, when Graph $\widetilde{G}$ is complete we have*

$$cut(A, B) = \frac{1}{2} \lim_{\delta_{\max} \to 0} \psi_\delta(c) \tag{11}$$

Therefore, the well-known kernel form of cut corresponds to the upper bound for the misclassification rate on the degraded δ-cover of the data points, i.e. $\delta_{\max} \to 0$, in case of fixed bandwidth and complete graph. In this special case the decision regions comprises only the discrete data points. We prefer the more general PKD term ψ_δ to cut, since ψ_δ is the upper bound for the more accurate misclassification rate. We use nearest neighbor δ-cover in practice, whose advantage over the degraded δ-cover is shown in experiments.

The Objective Function for Clustering

The PKD term ψ_δ achieves its minimum 0 by grouping all the data into a single cluster. To avoid such imbalanced partition, we introduce the sum of within-cluster dissimilarities term to control the size of clusters, and the cluster indicators enable us to express it conveniently, i.e. $\sum_{i=1}^{N} \|x_i - x_{c_i}\|^2$. For numerical stability, we take its Gaussian form, that is

$$u(c) = \sum_{i=1}^{N} u_i(c) = \sum_{i=1}^{N} 1 - \exp\left(-\|x_i - x_{c_i}\|^2\right) \tag{12}$$

so that u_i falls into $[0, 1]$. Moreover, it is important for the exemplar based clustering to ensure the consistency of the configuration of the resultant cluster indicators (Frey and Dueck 2007). That is, a data point should be the cluster exemplar of itself if it is taken as a cluster exemplar by another data point, and formally as below:

A configuration of the cluster indicators $c = \{c_i\}_{i=1}^N$ of data X is consistent iff $c_j = j$ when $c_i = j$ for any $i, j \in 1..N$.

Combining (12) and (9), the final objective function for PEC is defined as

$$E(c) = u(c) + \lambda(\psi_\delta(c) + \rho(c)) \tag{13}$$

where

$$\psi_\delta(c) = \sum_{i<j} \psi_{ij}^{\delta}(c_i, c_j) \tag{14}$$

,

$$\rho(c) = \sum_{i<j} \rho_{ij}(c_i, c_j)$$

$$\rho_{ij}(c_i, c_j) = \begin{cases} \infty & c_i = j, c_j \neq j \text{ or } c_j = i, c_i \neq i \\ 0 & \text{otherwise} \end{cases}$$

We rewrite the PKD term ψ_δ in a pairwise form in (14). Our goal is to find the optimal configuration of the cluster indicators, i.e. $c^* = \arg\min_c E(c)$. There are two terms in the objective function: the pairwise term $\psi_\delta(c)$ that represents the upper bound for the misclassification rate of the data partition corresponding to c, and the unary term $u(c)$ that controls the cluster size. The two terms are competing in the sense that $\psi_\delta(c)$ tends to make all data points group together while $u(c)$ encourages each data point form its own cluster, and λ is a balancing parameter. Therefore, the minimization of (13) searches for the best trade off between the two terms and adaptively determines the number of clusters in a model selection manner.

Optimization by Accelerated Belief Propagation

Due to the form of (13), it is straightforward to construct a pairwise Markov Random Field (MRF) representing the unary and pairwise terms as the data likelihood and prior respectively. The variables c are modeled as nodes and the unary term and pairwise term in the objective function are modeled as potential functions in the pairwise MRF. The minimization of the objective function is then converted to a MAP (Maximum a Posterior) problem on the pairwise MRF. While there are various inference techniques on MRF such as Iterated Conditional Model (Besag 1986), Simulated Annealing (Kirkpatrick, Gelatt, and Vecchi 1983; Barnard 1989), and Graph Cut (Boykov, Veksler, and Zabih 2001; Kolmogorov and Zabih 2004), we choose Max-Product Belief Propagation (BP) (Weiss and Freeman 2001) due to its satisfactory empirical performance and the speedup of inference gained by the special form of the pairwise term of the objective function. Max-Product BP is known to produce an exact MAP solution when the graph is a tree, and it achieves satisfactory empirical results on graph with loops (Sun, Zheng, and Shum 2003; Felzenszwalb and Huttenlocher 2006).

The max-product belief propagation maximizes the posterior in two steps:

Message Passing: It iteratively passes messages along each edge according to

$$m_{ij}^t(c_j) = \min_{c_i} \left(M_{ij}^{t-1}(c_i) + \psi_{ij}^{\delta}(c_i, c_j) + \rho_{ij}(c_i, c_j) \right) \tag{15}$$

$$M_{ij}^t(c_i) \triangleq \sum_{k \in N(i) \setminus j} m_{ki}^t(c_i) + u_i(c_i)$$

where m_{ij}^t is the message sent from node i to node j in iteration t, $N(i)$ is the set of neighbors of node i.

Obtaining the optimal label: After the message passing converges or the maximal number of iterations is achieved, the final belief for each node is

$$b_i(c_i) = \sum_{k \in N(i)} m_{ki}^T(c_i) + u_i(c_i)$$

And the resultant optimal c_i^* is $c_i^* = \arg\min_{c_i} b_i(c_i)$.

The speed bottleneck of Max-Product BP lies in the message computation in (15). The direct computation of $\left\{m_{ij}^t(c_j)\right\}_{c_j=1}^N$ requires $O\left(N^2\right)$ time complexity. Exploiting the sparsity of the pairwise term $(\psi_{ij}^{\delta}(c_i, c_j) + \rho_{ij}(c_i, c_j))$ on each edge (i, j), where both ψ_{ij}^{δ} and ρ_{ij} are binary-valued functions, we propose a new message computation algorithm to reduce the time complexity to $O(N)$ by Proposition 1. First we simplify (15) as below:

$$m_{ij}^t(c_j) = \begin{cases} \min_{c_i} \left(M_{ij}^{t-1}(c_i) + \psi_{ij}^{\delta}(c_i, j) \right) & c_j = j \\ M_{ij}^{t-1}(i) & c_j = i \\ \min_{c_i \neq j} \left(M_{ij}^{t-1}(c_i) + \psi_{ij}^{\delta}(c_i, c_j) \right) & c_j \neq i, j \end{cases} \tag{16}$$

Then we have

Proposition 1. *$\left\{m_{ij}^t(c_j)\right\}_{c_j=1}^N$ can be computed in $O(N)$ time for any fixed i, j, t*

Proposition 1 suggests an efficient message computation procedure in linear time, which reduces the time complexity of LBP from $O\left(TEN^2\right)$ to $O(TEN)$, where E is the number of edges in the pairwise MRF and T is the number of iterations of message passing. It significantly speeds up the inference in our graphical model.

Experimental Results

Default Parameter Setting

This section demonstrates the performance of PEC on synthetic and real data sets. We first introduce the default parameter setting for PEC. Note that the constructed pairwise MRF is a complete graph with $O\left(N^2\right)$ edges. Due to the Gaussian kernel for the weight of edges, we discard nearly 70 percents of the edges while retaining half of the total edge weight for each node without hurting the performance. The default value for h_0 in (2) is h_0^*, empirically set as the variance of the pairwise dissimilarities between data points $\left\{\|x_i - x_j\|_{i<j}\right\}$, and the default value for the balancing parameter λ in the objective function (13) is 1. We use these settings throughout all the experiments conducted.

Based on the objective function (13), increasing the balancing parameter λ or the initial fixed bandwidth h_0 for kernel density estimator will produce fewer clusters, and vice versa. Therefore, we can perform a series of model selection by varying both λ and h_0. We vary h_0 by $h_0 = \alpha h_0^*$,

Table 1: Real data sets used in experiments

	Iris	Wine	VC	BT
# of instances	150	178	310	106
Dimension	4	13	6	9
# of classes	3	3	3	6

Table 2: Clustering result on the first synthetic data

method	K-means	SC	GMM	AP	PEC
Avg. ARI	0.8307	0.7123	0.7847	0.6373	**0.8636**
SD	0.0560	0.0505	0.0621	0.0433	**0.0234**
AC	-	-	-	11.3	**5**

Table 3: Clustering result on the second synthetic data

method	K-means	SC	GMM	AP	PEC
Avg. ARI	0.8446	0.8710	0.9148	0.7548	**0.9461**
SD	0.0686	0.0564	0.0684	0.0992	**0.0377**
AC	-	-	-	8.6667	**5.0667**

where α, called the bandwidth ratio, is a parameter controlling the kernel bandwidth. In the model selection process λ varies between $[0, 1]$ and the bandwidth ratio α varies between $[0.2, 1.5]$ with step 0.1 and 0.05 respectively to produce different number of clusters, and this parameter setting is fixed for all the data sets in our experiments.

Data Sets

We conduct experiments on two synthetic data sets and four real data sets. For both synthetic data sets, we randomly generate 300 points in R^2 whose distribution is a mixture of 5 Gaussians with equal weight and different scales. In the first data set, The 5 components of the mixture Gaussians are $N\left((-6.5\ \ 0), 6\mathrm{I}\right)$, $N\left((5\ \ 5.5), 3\mathrm{I}\right)$, $N\left((5\ \ 0), 1.2\mathrm{I}\right)$, $N\left((5\ \ -5), \mathrm{I}\right)$ and $N\left((0\ \ 0), 1.2\mathrm{I}\right)$. The specified parameters of the Gaussian components render a scenario where the data exhibits large scale difference and the cluster centers are relatively close to each other. We repeat the simulation 10 times. In the second data set, the means and covariance matrices for the Gaussian components are randomly generated. The means for the 5 Gaussian components are generated from $N\left((-6\ \ -6), \mathrm{I}\right)$, $N\left((-6\ \ 6), \mathrm{I}\right)$, $N\left((6\ \ 6), \mathrm{I}\right)$, $N\left((6\ \ -6), \mathrm{I}\right)$ and $N\left((0\ \ 0), \mathrm{I}\right)$ respectively. The covariance matrices for the first four Gaussian components are generated from $W\left(\mathrm{I}, 2\right)$, and the last covariance matrix is generated from $W\left(2\mathrm{I}, 2\right)$, where $W\left(\Sigma, d\right)$ indicates Wishart distribution with covariance matrix Σ and d is the degree of freedom. We sample the means and covariance matrices for the 5 Gaussian components 3 times, and for each parameter setting of the 5 Gaussians we generate the data 5 times.

We choose four real data sets from UCI repository (A. Asuncion 2007), i.e. Iris, Wine, Vertebral Column (VC), and Breast Tissue (BT), which are summarized in Table 1.

Evaluation Metric

We use the popular adjusted rand index (ARI) (Hubert and Arabie 1985) to evaluate the performance of the clustering methods. ARI is the adjusted-for-chance version of rand index, and it has been widely used as a measure of agreement between two partitions. ARI varies from 1 for a perfect match to 0 for an entire random data partition, and a higher ARI indicates a better agreement between the partition obtained from clustering methods and the ground truth partition. Given a set of S data points and two partitions of these data points, i.e. $U = \{U_i\}_{i=1}^{K_1}$ and $V = \{V_j\}_{j=1}^{K_2}$, we denote the number of common objects of cluster U_i and V_j as n_{ij}, namely $n_{ij} = |U_i \cap V_j|$. Then ARI is defined as below:

$$\mathrm{ARI} = \frac{\mathrm{Index} - \mathrm{ExpectedIndex}}{\mathrm{MaxIndex} - \mathrm{ExpectedIndex}}$$

$$\mathrm{Index} = \sum_{i=1}^{K_1}\sum_{j=1}^{K_2}\binom{n_{ij}}{2}$$

$$\mathrm{ExpectedIndex} = \left[\sum_{i=1}^{K_1}\binom{|U_i|}{2}\cdot\sum_{j=1}^{K_2}\binom{|V_j|}{2}\right]\Big/\binom{S}{2}$$

$$\mathrm{MaxIndex} = \frac{1}{2}\left[\sum_{i=1}^{K_1}\binom{|U_i|}{2} + \sum_{j=1}^{K_2}\binom{|V_j|}{2}\right]$$

Clustering on Synthetic Data

We compare PEC to K-means, spectral clustering (SC) (Ng, Jordan, and Weiss 2001), Gaussian Mixture Model (GMM), and Affinity Propagation (AP) (Frey and Dueck 2007). We feed the ground truth number of clusters to K-means, SC and GMM. We require AP and PEC to do model selection only once with default parameter. The result for the two synthetic data sets are shown in Table 2 and Table 3 respectively, where Avg. ARI stands for average ARI and SD stands for standard deviation of the ARI, AC stands for average number of clusters by model selection. PEC actually chooses the correct number of clusters in all but one of the simulations, and we observe that it achieves the highest average ARI. On contrast AP tends to split data into a larger number clusters. Although GMM makes a correct assumption about the underlying distribution of the data, PEC is better than GMM in both performance stability and average ARI, by modeling the multiscale data more accurately through the variable bandwidth kernel density estimator.

Clustering on Real Data

We compare PEC to K-means, Gaussian Mixture Models and spectral clustering (Ng, Jordan, and Weiss 2001) on the four UCI data sets. The clustering results are shown in Figure 2, 3, 4 respectively. We plot the performance of the clustering method versus the number of clusters. Since spectral clustering is dependent on the kernel bandwidth, we run it with two bandwidth choices. The first choice is the default empirical one, where the bandwidth is set as 0.05 times the maximal pairwise dissimilarities between data points. The second choice is to set the bandwidth the same as h_0, which varies with respect to the number of clusters for the sake

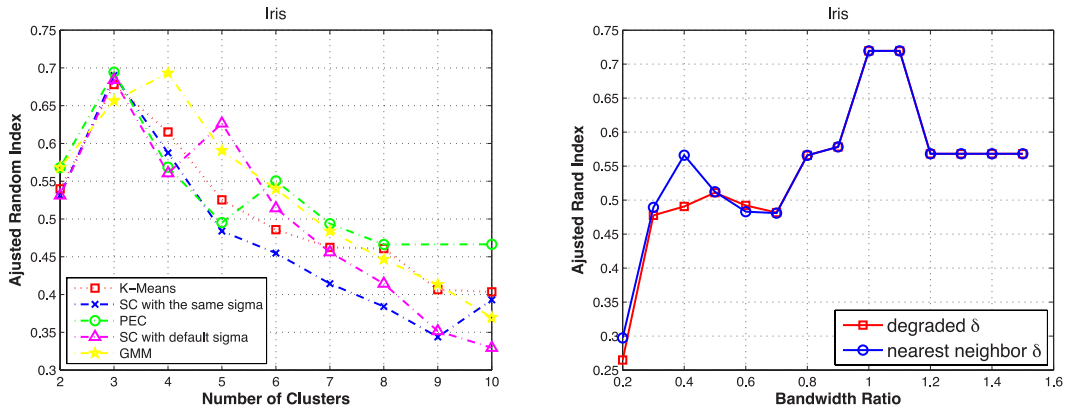

Figure 2: Clustering on UCI Iris data set and the comparison between nearest neighbor δ-cover and the degraded δ-cover

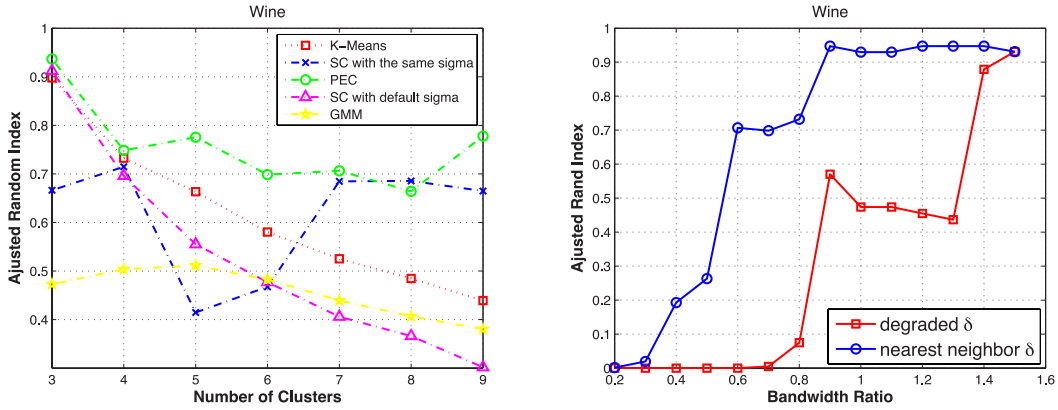

Figure 3: Clustering on UCI Wine data set and the comparison between nearest neighbor δ-cover and the degraded δ-cover

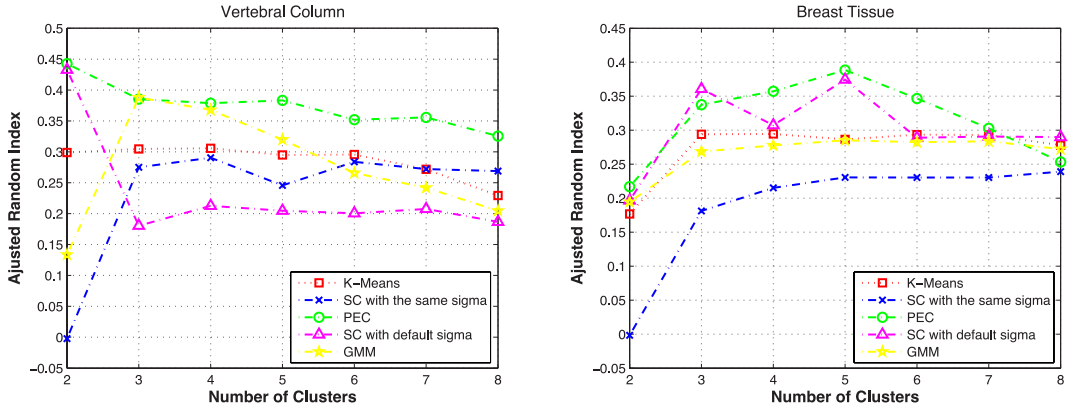

Figure 4: Clustering on UCI Vertebral Column and Breast Tissue data sets

of a fair comparison between PEC and spectral clustering. We normalize the Wine and BT data (i.e. make each attribute have unit variance) since some attributes have much larger variance than other attributes in the two data sets. For the clustering methods that depend on random initialization (such as K-means), we run them 30 times and take the average performance. We observe that PEC constantly outperforms K-means, GMM and spectral clustering in case of

Table 4: Comparison between AP and PEC

Data sets	Iris	Wine	VC	BT
AP	**0.7296**	0.7328	0.2937	0.2691
Preference	-51.6843	-281.4668	-104320	-40.2999
PEC	0.6943	**0.9366**	**0.3854**	**0.3465**

Wine, VC and BT data sets. Note that PEC not only renders comparable or better result when it chooses the ground truth number of clusters, it also produces satisfactory ARI scores over a range of the cluster numbers for all the four data sets. This demonstrate the effectiveness of our new measure, i.e. the misclassification rate of data partition, in sensibly separating the clusters and the ability of the employed kernel density estimator to model the underlying data distributions.

Affinity Propagation (AP) controls the number of clusters by a parameter called preference, and there is little theoretical justification on the setting of the preference (Tarlow, Zemel, and Frey 2008). The preference of AP needs to be tuned separately for each data set, and AP does not offer a way of generating different number of clusters by varying its preference value over a small fixed range. In contrast, the parameters of PEC are not sensitive to different data sets and they vary within a relatively small and fixed range for all the four data sets. In order to render a fair comparison we show the clustering ARI of AP when its preference value is tuned to produce the correct number of clusters in Table 4. For each data set, we first estimate the lower bound and upper bound for the preference (AP chooses 1 or 2 clusters and $N-1$ or N clusters for such lower bound and upper bound for its preference respectively), then we evenly sample 234 (the number of times PEC performs model selection) preference values between its upper bound and lower bound, and run AP with the sampled preference values. We record the average performance and the average preference value of AP when it chooses the correct number of clusters. We observe that PEC still behaves favorably to AP for the Wine, VC and BT data sets, and the preference of AP that generates correct number of clusters changes significantly across different data sets.

For Iris and Wine we further show the performance comparison between nearest neighbor δ-cover and the degraded δ-cover in the same framework of optimizing the objective function (13), with respect to the bandwidth ratio. We observe that nearest neighbor δ-cover often improves the performance compared to the degraded δ-cover, especially in the Wine data set, and it never significantly hurts the performance. We attribute this improvement to the fact that the nearest neighbor δ-cover enables the PKD term ψ_δ to approximate the true misclassification rate of the data partition more accurately.

Conclusion

We propose a new clustering method, Pairwise Exemplar Clustering (PEC), to incorporate kernel methods into an exemplar-based clustering scheme. PEC employs kernel density estimation to model the underlying data distributions, and utilizes a new measure, i.e. misclassification rate of data partition, to well separate clusters. An objective function is built based on the new measure, and the number of clusters is determined by optimizing the objective function through efficient message computation in a Pairwise MRF. Experimental results show the effectiveness of our method on various data sets.

Acknowledgement

This research is supported in part by ONR Grant N000141210122; and in part by Google Faculty Research Award.

Appendix

Proof of Theorem 1.

$$\begin{aligned}
\widetilde{P}_\delta(M) &= \sum_{j=1}^{Q} \int_{\bigcup_{i\neq j} \mathcal{R}_i^\delta} p(x|C_j)\,\pi_j dx \\
&= \sum_{j=1}^{Q} \int_{\bigcup_{i\neq j} \mathcal{R}_i^\delta} f_j(x)\,\pi_j dx \\
&= \sum_{j=1}^{Q} \int_{\bigcup_{i\neq j} \mathcal{R}_i^\delta} \frac{1}{|C_j|} \sum_{l=1}^{N} \frac{1}{h_l^D} K\left(\frac{x-x_l}{h_l}\right) \mathcal{I}_{C_j}(x_l) \cdot \frac{|C_j|}{N} dx \\
&= \frac{1}{N} \sum_{j=1}^{Q} \sum_{x_m \notin C_j} \sum_{x_l \in C_j} \int_{B(x_m,\delta_m)} \frac{1}{h_l^D} K\left(\frac{x-x_l}{h_l}\right) dx \\
&= \frac{1}{N} \sum_{m=1}^{N} \sum_{l=1}^{N} \theta_{lm} \int_{B(x_m,\delta_m)} \frac{1}{h_l^D} K\left(\frac{x-x_l}{h_l}\right) dx \qquad (17)
\end{aligned}$$

Due to the nonnegativity and differentiability of K, we perform first order Taylor expansion for K and there exist a $x_m^* \in B(x_m, \delta_m)$ such that

$$\begin{aligned}
K\left(\frac{x-x_l}{h_l}\right) &= \left| K\left(\frac{x-x_l}{h_l}\right) \right| \\
&\leq K\left(\frac{x_m-x_l}{h_l}\right) + \frac{1}{h_l} \left\| \nabla K\left(\frac{x_m^*-x_l}{h_l}\right) \right\| \|(x-x_m)\| \\
&\leq K\left(\frac{x_m-x_l}{h_l}\right) + \frac{G_{ml}\delta_m}{h_l}
\end{aligned}$$

and it follows that

$$\begin{aligned}
&\int_{B(x_m,\delta_m)} \frac{1}{h_l^D} K\left(\frac{x-x_l}{h_l}\right) dx \\
&\leq \frac{c_0 \delta_m^D}{h_l^D} \left(K\left(\frac{x_m-x_l}{h_l}\right) + \frac{G_{ml}\delta_m}{h_l} \right) \qquad (18)
\end{aligned}$$

where $c_0\delta_m^D$ is the volume of the ball with radius δ_m in D dimensional space. Substitute (18) into (17), we have

$$\begin{aligned}
\widetilde{P}_\delta(M) &= \frac{1}{N} \sum_{m=1}^{N} \sum_{l=1}^{N} \theta_{lm} \int_{B(x_m,\delta_m)} \frac{1}{h_l^D} K\left(\frac{x-x_l}{h_l}\right) dx \\
&\leq \frac{c_0}{N} \sum_{m=1}^{N} \sum_{l=1}^{N} \theta_{lm} \frac{\delta_m^D}{h_l^D} \left(K\left(\frac{x_m-x_l}{h_l}\right) + \frac{G_{ml}\delta_m}{h_l} \right) \\
&\leq \frac{c_0}{N} \left(\frac{\delta_{\max}}{h_{\min}}\right)^D \psi_\delta(c)
\end{aligned}$$

□

Proof of Proposition 1: The computation of $m_{ij}^t(j)$ costs $O(N)$ time. For all $c_j \neq j$, we compute

$$m_{ij}^t(c_j) = \min_{c_i \neq j} \left(M_{ij}^{t-1}(c_i) + \psi_{ij}^\delta(c_i, c_j) \right), c_j \neq j \quad (19)$$

Since $\psi_{ij}^\delta(c_i, c_j) \propto \theta_{ij}$ is a Potts model potential function, and both c_i, c_j range over a common label set $L \triangleq \{1, 2, ...N\} \setminus \{j\}$, it is shown in (Felzenszwalb and Huttenlocher 2006) that $\left\{m_{ij}^t(c_j)\right\}_{c_j \in L}$ can be computed in $O(N)$ time by distance transform. After that, all $\left\{m_{ij}^t(c_j)\right\}_{c_j \in L}$ are correct except $m_{ij}^t(i)$ according to (16). We then recompute $m_{ij}^t(i)$ by (16), which also requires $O(N)$ time. Therefore the entire computation of $\left\{m_{ij}^t(c_j)\right\}_{c_j=1}^N$ costs $O(N)$ time. □

References

A. Asuncion, D. N. 2007. UCI machine learning repository.

Abramson, I. S. 1982. On bandwidth variation in kernel estimates-a square root law. *The Annals of Statistics* 10(4):pp. 1217–1223.

Barnard, S. T. 1989. Stochastic stereo matching over scale. *International Journal of Computer Vision* 3(1):17–32.

Besag, J. 1986. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society* B-48:259–302.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

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

Comaniciu, D., and Meer, P. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* 24(5):603–619.

Felzenszwalb, P. F., and Huttenlocher, D. P. 2006. Efficient belief propagation for early vision. *International Journal of Computer Vision* 70(1):41–54.

Fraley, C., and Raftery, A. E. 2002. Model-Based Clustering, Discriminant Analysis, and Density Estimation. *Journal of the American Statistical Association* 97(458):611–631.

Frey, B. J., and Dueck, D. 2007. Clustering by passing messages between data points. *Science* 315:972–977.

Gomes, R.; Krause, A.; and Perona, P. 2010. Discriminative clustering by regularized information maximization. In *NIPS*, 775–783.

Hinneburg, A., and Gabriel, H.-H. 2007. Denclue 2.0: Fast clustering based on kernel density estimation. In *IDA*, 70–80.

Hubert, L., and Arabie, P. 1985. Comparing Partitions. *Journal of Classification*.

Kirkpatrick, S.; Gelatt, C. D.; and Vecchi, M. P. 1983. Optimization by Simulated Annealing. *Science, Number 4598, 13 May 1983* 220, 4598:671–680.

Kolmogorov, V., and Zabih, R. 2004. What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Anal. Mach. Intell.* 26(2):147–159.

Lashkari, D., and Golland, P. 2007. Convex clustering with exemplar-based models. In *NIPS*.

Li, J.; Ray, S.; and Lindsay, B. G. 2007. A nonparametric statistical approach to clustering via mode identification. *J. Mach. Learn. Res.* 8:1687–1723.

Mclachlan, G. J. 2004. *Discriminant Analysis and Statistical Pattern Recognition (Wiley Series in Probability and Statistics)*. Wiley-Interscience.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. In *NIPS*, 849–856.

Silverman, B. W. 1986. *Density Estimation for Statistics and Data Analysis*, volume 37. Chapman and Hall.

Sun, J.; Zheng, N.; and Shum, H.-Y. 2003. Stereo matching using belief propagation. *IEEE Trans. Pattern Anal. Mach. Intell.* 25(7):787–800.

Takahashi, R. 2011. Sequential minimal optimization in adaptive-bandwidth convex clustering. In *SDM*, 896–907.

Tarlow, D.; Zemel, R. S.; and Frey, B. J. 2008. Flexible priors for exemplar-based clustering. In *UAI*, 537–545.

Terrell, G. R., and Scott, D. W. 1992. Variable Kernel Density Estimation. *The Annals of Statistics* 20(3):1236–1265.

Weiss, Y., and Freeman, W. T. 2001. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory* 47(2):736–744.

Wu, Z., and Leahy, R. M. 1993. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 15(11):1101–1113.

Xu, R., and II, D. C. W. 2005. Survey of clustering algorithms. *IEEE Transactions on Neural Networks* 16(3):645–678.

Xu, L.; Neufeld, J.; Larson, B.; and Schuurmans, D. 2004. Maximum margin clustering. In *NIPS*.