Unsupervised distance metric learning using predictability

Abhishek A. Gupta
Department of Statistics
University of Pennsylvania
abgupta@wharton.upenn.edu

Dean P. Foster
Department of Statistics
University of Pennsylvania
foster@wharton.upenn.edu

Lyle H. Ungar
Department of Computer and Information Science
University of Pennsylvania
ungar@cis.upenn.edu

Abstract

Distance-based learning methods, like clustering and SVMs, are dependent on good distance metrics. This paper does *unsupervised* metric learning in the context of clustering. We seek transformations of data which give *clean* and well separated clusters where *clean* clusters are those for which membership can be accurately predicted. The transformation (hence distance metric) is obtained by minimizing the *blur ratio*, which is defined as the ratio of the within cluster variance divided by the total data variance in the transformed space. For minimization we propose an iterative procedure, *Clustering Predictions of Cluster Membership* (CPCM). CPCM alternately (a) predicts cluster memberships (e.g., using linear regression) and (b) clusters these predictions (e.g., using k-means). With linear regression and k-means, this algorithm is guaranteed to converge to a fixed point. The resulting clusters are invariant to linear transformations of original features, and tend to eliminate noise features by driving their weights to zero.

1 Introduction

In data mining one often wants to find *good* clusters in a set of data using e.g. k-means, agglomerative or spectral clustering methods (Jain et al., 1999; Ng et al., 2002). Though it is clear that clusters depend on the distance metric, what constitutes a *good* cluster is a difficult question, and depends on the goal of the clustering. Many probabilistic, information theoretic, and graph theoretic measures have been proposed to capture the quality of clustering (Kannan et al., 2000). Most of these criteria try to measure the similarity between points in a cluster, and thus depend on some distance metric. In a supervised setting, such as for classification problems, researchers have developed criteria for learning the distance metrics, either when all points are labeled with clusters (Shalev-Shwartz et al., 2004; Shental et al., 2002), or when "side information" about pairs of points either being, or not being in the same cluster (Xing et al., 2003) are available.

We propose a method of learning metrics in an unsupervised setting. A good distance metric would lead to tight and well-separated clusters in some projected space. We quantify this by introducing a new criterion, the ratio of the average distance of points to their nearest cluster centers to the average distance of the data points to their overall mean in the transformed space. For a linear transformation $\mathbf{A}$ of the data, we call our criterion the *blur ratio* $BR(\mathbf{A})$. Our goal then is to find the $\mathbf{A}$ which minimizes this *blur ratio*. The criterion resembles the one that *Linear Discriminant Analysis* (LDA) minimizes, except that we are doing unsupervised learning, while LDA assumes that labels are known. In effect, we are learning a distance metric such that the transformation obtained projects the data to a subspace where the clusters are tight. Minimization of the *blur ratio* handles data with high-dimensional noise features by tending to drive their weights to zero.

For the minimization we propose an iterative algorithm, *Clustering Predictions of Cluster Membership* (CPCM), which first predicts cluster membership, and then defines new clusters by clustering the predictions of cluster membership. In CPCM we combine a hard clustering algorithm with a soft prediction algorithm (i.e. the prediction step predicts cluster probabilities and not cluster memberships). The intuition behind using predictability is that if we generate clusters using a set of features, we should also be able to predict the membership of the clusters using the same features. By using predictions of cluster membership, we can take advantage of supervised learning methods to better solve unsupervised and semi-supervised problems (Chapelle et al., 2003). This paper explains the CPCM algorithm in detail and characterizes some of its properties for the case where linear regression is used for prediction, including showing that it gives clusters which are invariant to linear transformations of the data. CPCM can be easily extended to different prediction and clustering techniques; we discuss the use of Reproducing Kernel Hilbert Spaces (RKHS) for prediction in this context, and show that CPCM gives superior perfomance to similar metric learning and unsupervised methods when there are many spurious features. CPCM reduces the contribution of irrelevant features, greatly improving cluster quality.

2 Optimal Distance Metric and Blur Ratio

Consider a $N \times p$ dimensional matrix $\mathbf{X}$ where N is the number of points and p is the number of features. We will use $\mathbf{X}_i$ to denote the ith row and $\mathbf{X}_{\cdot i}$ to denote the ith column in $\mathbf{X}$. This makes the feature vector into a row vector. We will use row vectors throughout. Define $\mathbb{C} = \{C_1, \cdots, C_K\}$ to be the set of clusters. Let $\triangle_{\mathbb{C}}$ be the simplex over the K dimensions (the subspace of K dimensions such that every point looks like a probability, with components lying in (0,1) and summing to 1) with $e_k \in \triangle_{\mathbb{C}}$ denoting a unit direction. Denoting the clustering function by c, we have the following map

$$\mathbf{X} \overset{k\text{-means}}{\longrightarrow} c(\,) : \mathcal{R}^p \rightarrow \mathbb{C} \tag{1}$$

Using $c()$, we define the matrix $\mathbf{Z}$ such that $\mathbf{Z}_i = e_{k:c(\mathbf{X}_i)=C_k}$. Note that $\mathbf{Z}_i$ is a K dimensional row vector on the simplex $\triangle_{\mathbb{C}}$. $\triangle_{\mathbb{C}}$ can be regarded as a probability simplex for the cluster membership with $\mathbf{Z}_{ik}$ equaling the probability of point i being in cluster k. Define $\boldsymbol{\mu}_k \in \mathcal{R}^p$ as the center in the feature space for cluster C_k.

Our goal is to find the linear transformation of the data $\mathbf{X}$ such that the distance metric $d(x, y) = \sqrt{(x-y)\mathbf{A}(x-y)^T}$ gives the lowest *blur ratio*. We will build up the *blur ratio* by a sums of squares decomposition: Within cluster variance, $SSC \equiv \displaystyle\sum_{k=1}^{K} \sum_{i:c(\mathbf{X}_i)=C_k} (\mathbf{X}_i - \boldsymbol{\mu}_k)\mathbf{A}(\mathbf{X}_i - \boldsymbol{\mu}_k)^T$ and, Total variance, $SST \equiv \displaystyle\sum_{i=1}^{N} (\mathbf{X}_i - \boldsymbol{\mu})\mathbf{A}(\mathbf{X}_i - \boldsymbol{\mu})^T$. Here $\boldsymbol{\mu} = \bar{\mathbf{X}}$ and $\mathbf{A}$ is a symmetric positive semi-definite matrix. We can then define the *blur ratio* and the optimization problem as

$$\min_{\mathbf{A}, c} BR(\mathbf{A}, c) \equiv \frac{SSC}{SST}$$

Following the argument for k-means type algorithms (Peng & Xia, 2005), it is clear that optimizing the *blur ratio* is NP-hard. Thus we rely on the existence of good approximate clustering algorithms. Given the cluster partition, the optimum $\mathbf{A}$ matrix unfortunately will be of rank one (a similar property was pointed out by Xing et al., 2003) Instead we want to ensure that the transformation minimizes the distance between cluster centers and a K-dimensional simplex, while maintaining the simplex structure. We therefore add the following constraint $\quad (\forall i \neq j) \quad (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)\mathbf{A}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T = 2$ which prevents the centers of two different clusters from overlapping, and keeps the $\mathbf{A}$ matrix from collapsing to a rank one matrix. Note that the RHS of the constraint just has to be any positive number (which we choose as 2). Without loss of generality, under this constraint, we can take that there exists a decomposition of $\mathbf{A} = \boldsymbol{\beta}\boldsymbol{\beta}^T$ (since $\mathbf{A}$ is positive definite) so that,

Algorithm 1 CPCM

```
Input: Data $\mathbf{X}$. Set $t$=0
Generate an initial set of random clusters $\mathbb{C}^{(0)}$
repeat
    Predict cluster membership $\mathbf{Z}_i^{(t)}$ based on $\mathbf{X}$ and $\mathbb{C}^{(t)}$
    Generate $\mathbb{C}^{(t+1)}$ by clustering $\widehat{\mathbf{Z}}^{(t)}$ and increment $t$.
until $BlurRatio(t)$ = $BlurRatio(t-1)+\epsilon$.
```

$SSC = \sum_{k=1}^{K} \sum_{i:c(i)=C_k} (\mathbf{X}_i\boldsymbol{\beta}-\boldsymbol{\theta}_k)(\mathbf{X}_i\boldsymbol{\beta}-\boldsymbol{\theta}_k)^T$. Thus given the clustering, the SSC can be minimized by minimizing the L_2 distance shown above, which boils down to finding the optimal $\boldsymbol{\beta}$ and hence $\mathbf{A}$. Thus we can consider $\boldsymbol{\theta}_k \in \triangle_{\mathbb{C}} \subset \mathcal{R}^K$ as the center in the prediction space for cluster C_k.

Finding the optimal BR can thus be seen as a two stage optimization procedure. We first find an optimal partitioning of the data using a clustering method. Then, we find the optimal $\mathbf{A}$ which reduces the within cluster variance in the transformed space. These two steps can be iterated till a fixed point. The basic intuition behind a two step procedure is that the points which cluster together lie around the corners of a simplex. In CPCM we use a hard clustering algorithm and a soft prediction algorithm. The soft prediction from linear regression finds the probabilities of cluster memberships and separates the clusters based on features. Hard clustering then drives the transformed points towards the simplex corners.[1] Armed with this insight, we now introduce the CPCM algorithm.

3 CPCM: An Iterative Clustering Prediction algorithm

The CPCM algorithm alternates between two stages - Clustering and Prediction. In this paper we use linear regression as the prediction algorithm, and k-means as the clustering algorithm. More formally, the cluster prediction model is given by: $\mathbf{Z}_i = \mathbf{X}_i\boldsymbol{\beta} + noise$, where $\boldsymbol{\beta} \in \mathcal{R}^{p\times K}$. Estimating the cluster memberships of each point by least squares gives a prediction $\widehat{\mathbf{Z}}_i$. Since $\widehat{\mathbf{Z}}_i \in \mathcal{R}^K$ (instead of just a scalar as in the case of a usual regression setting), we run a regression on each of the K columns of $\mathbf{Z}$ ($N \times K$ matrix) to generate the predictions. The points are then clustered in the cluster-prediction space, which is a linear transformation of the original points: $\widehat{\mathbf{Z}} \overset{\text{hard clustering algo.}}{\longrightarrow} c'(\ ) : \mathcal{R}^K \to \mathbb{C}'$. The algorithm is then iterated.

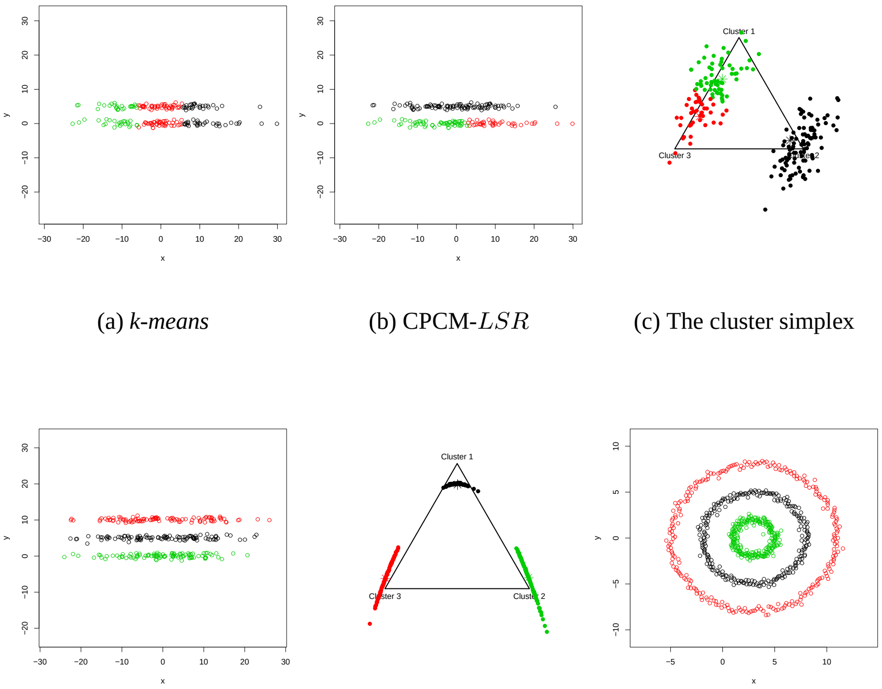
Figure 1 shows an example dataset consisting of a blurry version of two line segments. When k-means was used for clustering, the clusters produced failed to capture the structure of the data; since k-means tries to minimize the within-cluster variance parallel bands are produced (Figure 1(a)). In contrast, CPCM finds clusters that better capture the structure (Figure 1(b)). Figure 1(c) shows the projection of the points in the simplex space, which is a linear transformation of the original data. Key to CPCM is the fact that it clusters in a simplex space. We can view these K clusters in a $K-1$-simplex $\triangle_{\mathbb{C}}$. The predicted cluster memberships can then be viewed as probabilities.

3.1 CPCM computes a (local) minimum of the *blur ratio*

Recall that $\boldsymbol{\mu}_{(\cdot)}$ live in the feature space and $\boldsymbol{\theta}_{(\cdot)}$, are their projections in the simplex space. When using k-means, $\mathbf{Z}_i$ are at the corners of the simplex, $\triangle_{\mathbb{C}}$ and so $\mathbf{Z}_i = e_k$ when point i is in cluster C_k. Also, we have $\mathbf{X}_i \in \mathcal{R}^p$. As discussed above, CPCM works in two stages:

1. **Prediction Stage:** Given cluster memberships $\mathbf{Z}_i$, we minimize $RSS_P(\boldsymbol{\beta}) = \sum_{i=1}^N ||\mathbf{Z}_i - \mathbf{X}_i\boldsymbol{\beta}||^2$ for $\boldsymbol{\beta} \in \mathcal{R}^{p\times K}$. The solution $\mathbf{X}_i\widehat{\boldsymbol{\beta}} \in \triangle_{\mathbb{C}}$ gives the predicted location of the points in the cluster probability simplex. When LSR is used with an intercept, the predicted probabilities automatically sum to one for each point (Hastie et al., 2001).

[1] Alternatively, one could combine a hard prediction with a soft clustering algorithm where the hard regression drives the transformed points towards the simplex corners. Using a hard clustering and hard prediction or soft clustering and soft prediction would yield a trivial fixed point.



(a) *k-means* (b) CPCM-LSR (c) The cluster simplex

(d) Three parallel clouds (e) The cluster simplex (f) Three concentric circles

Figure 1: An example run is shown from (a)-(c). Data is from two parallel lines with Gaussian noise. (a) K-means creates parallel vertical bands to reduce the within cluster variance, and fails to cluster "correctly." *Blur ratio* for $K = 3$, $BR_3 = 0.26$ and for $K = 2$, $BR_2 = 0.447$ (b) In contrast, CPCM finds the underlying elongated band structure. *Blur ratio* for $K = 3$, $BR_3 = 0.148$ and for $K = 2$, $BR_2 = 0.041$. Thus, *blur ratio* is indeed minimized by CPCM for the right number of clusters. (c) Linear regression transforms the data linearly on to the simplex. Examples using CPCM-*RKHS* with a Gaussian kernel are shown from (d)-(f). (d) Three parallel lines (e) Projection on the simplex space shows that CPCM tries to pull the cluster centers to the simplex corners. (f) Three concentric circles.

2. **Clustering Stage:** Now minimize the objective function for the k-means algorithm, $RSS_C(\boldsymbol{\mu}, c(\cdot)) = \sum_{i=1}^{N} ||\mathbf{X}_i\boldsymbol{\beta} - \boldsymbol{\mu}_k\boldsymbol{\beta}||^2$. Clearly $\boldsymbol{\mu}_k$ is only identified for the subspace spanned by $\boldsymbol{\beta}$. Effectively, we are taking the distance orthogonal to the $\boldsymbol{\beta}$ subspace as having zero distance. To make this clearer, $\boldsymbol{\theta}_k \equiv \boldsymbol{\mu}_k\boldsymbol{\beta} \in \mathcal{R}^K$. The clustering function then is basically, $c(\mathbf{X}_i) = \underset{1\leq k\leq K}{\text{argmin}} \; ||\mathbf{X}_i\boldsymbol{\beta} - \boldsymbol{\theta}_k||^2$ where, $\boldsymbol{\theta}_k \equiv (\mathbf{X}_i/|C_k|)\boldsymbol{\beta} = \boldsymbol{\mu}_k\boldsymbol{\beta}$.

3.2 Properties of the fixed point

We now state and prove two guarantees about CPCM: it gives a result which is invariant to linear transformations of the original space, and it always converges.

Property 1 (Linear Transformation Invariance) *The algorithm is invariant to linear transformations of the data.*

Proof The data, $\mathbf{X_i}$ is only used directly by the linear regression step; clustering only uses the projected $\mathbf{X_i}\boldsymbol{\beta}$. Since the linear regression estimates a β which is invariant to linear transformation, the whole algorithm is invariant to linear transformations. Note that if regularization is used, it will enforce some structure on the initial data. Hence we will lose this particular invariance property. This regularization is necessary in the RKHS setting.

Since the initial cluster labels are chosen at random, there are a multitude of possible ending labels. Thus we need to know that regardless of the initial state, that the algorithm will converge. We define the set of possible convergent points as $\mathcal{F}$:

Definition 1 (Fixed Point) *Define the set of sets of points, $\mathcal{F}$, as the **fixed point** if $\mathcal{F} = \mathbb{C}^{(j)} = \mathbb{C}^{(j+1)}$. Thus at the fixed point the clusters remain invariant after successive iterations.*

Property 2 (Existence of Fixed Point) *For any initial starting assignment of labels, the algorithm converges to a fixed point, $\mathcal{F}$ in finite time.*

Proof We first show that the algorithm converges to a local minimum of the *blur ratio* and then show that this local minimum corresponds to a fixed point as defined above. Since we update only if the *blur ratio* decreases, the iterations monotonically decrease the *blur ratio*. Hence, the algorithm stops when a local minimum of the *blur ratio* is reached. A trivial proof of the second part relies on there only being a finite number of possible splits into clusters. Hence due to the monotonicity, it must find a local minimum.

These two properties together show that the set of fixed points, $\mathcal{F}$, is invariant to linear transformations in $\mathbf{X}$. Also since k-means is $O(Np)$ and LSR is $O(Np^2)$, convergence is usually fast.

3.3 Extensions using RKHS

Whenever the clusters are linearly separable, the prediction step using linear regression is able to find the clusters. But, when the original feature space is not linearly separable, we will need a non-linear transformation of the features, say by using a Reproducing Kernel Hilbert Space (RKHS) (Hastie et al., 2001) for the prediction step. CPCM works well on the prototypical datasets considered in spectral clustering literature (Ng et al., 2002). Figure 1(d-f) show examples using an RKHS, where CPCM is able to correctly identify the three parallel bands and three concentric circles. Predicting with an RKHS can be viewed as transforming the data by a non-linear map to a linearly separable feature space and then applying CPCM in this image space.

4 Related Work

4.1 Unsupervised Learning

One of the most popular clustering techniques, k-means, minimizes the *blur ratio* with the $\mathbf{A} = \mathbf{I}$. Another, related, approach to clustering is to use a model-based clustering, e.g, Gaussian Mixture Models (GMM) (Dasgupta, 1999). GMMs with a single covariance also minimize the *blur ratio*, as they cluster in a linearly transformed space.

Another alternative to the prediction step of the blur ratio optimization would be to reduce to a generalized eigenvalue problem. Given a partition $\mathbb{C}$, the *blur ratio* can be considered as $BR(\mathbf{A} = \boldsymbol{\beta}\boldsymbol{\beta}^T, c()) \equiv \frac{tr(\mathbf{X}^{(c)}\mathbf{A}\mathbf{X}^{(c)T})}{tr(\mathbf{X}^{(m)}\mathbf{A}\mathbf{X}^{(m)T})} = \frac{tr(\boldsymbol{\beta}^T\mathbf{S}_c\boldsymbol{\beta})}{tr(\boldsymbol{\beta}^T\mathbf{S}_m\boldsymbol{\beta})}$ where $\mathbf{X}^{(c)}$ is the matrix with ith row as $\mathbf{X}_i - \boldsymbol{\mu}_{c(\mathbf{X}_i)}$, $\mathbf{X}^{(m)}$ is the matrix with ith row as $\mathbf{X}_i - \boldsymbol{\mu}$, $\mathbf{S}_c = \mathbf{X}^{(c)T}\mathbf{X}^{(c)}$ and $\mathbf{S}_m = \mathbf{X}^{(m)T}\mathbf{X}^{(m)}$. The $\boldsymbol{\beta}_i$ are solutions to the generalized eigenvalue problem $\mathbf{S}_c\beta_i = \lambda\mathbf{S}_m\beta_i$. The minimum *blur ratio* is obtained by taking the minimum eigenvalue, and taking all other directions zero (and hence $\boldsymbol{\beta}$ is rank 1). To avoid projection to a line, we constrain the minimum to be a projection onto K-1 dimensions. Optimizing this ratio is equivalent to CPCM with Linear Discriminant Analysis (LDA, with K classes) as the prediction step (Fukunaga, 1990, Hastie et al., 2001).

Recently we have been made aware of similar work done by Ding and Li (2007) who introduce LDA-km. This is an extension of the adaptive dimension reduction methodology (Ding et al., 2002), where they introduce ADR-EM. The ADR framework aims to do subspace clustering and uses cluster membership to define a K-1 dim space, but it still uses projections based on Euclidean distances. LDA-km, on the other hand, minimizes a criterion related to *blur-ratio*, and is similar to the CPCM-LDA suggested above. LDA and multiple regression are closely related techniques; See Hastie et al.

(1994) and Hastie et al. (1995) for a detailed discussion on the similarity and differences between the two. Key advantage of CPCM is that we can extend it to different prediction techniques e.g.to RKHS above, and to stepwise in next section. We compare LDA-km to CPCM in section 5.

Almost none of the traditional clustering and dimensionality reduction methods are invariant to linear transformations of the data (Frey & Jojic, 2003; Kumar & Orlin, 2005). For example, k-means requires a distance function which can be changed drastically by doing a linear transformation of the features. Different transformations lead to different clusters. One way to obtain invariant clusters is to use a clustering criterion which itself is invariant. This is the approach taken by Friedman and Rubin (1967) where they use the Mahalanobis distance. The circular k-means (Charalampidis, 2005) which is rotation invariant also utilizes the Mahalanobis distance in its criteria. Other related work includes Fitzgibbon and Zisserman (2002), Kumar and Orlin (2005), and Frey and Jojic (2003). Though invariant, none of these procedures learn metrics.

4.2 Distance Metric Learning

Several supervised learning methods, including the Mahalanobis Metric for Clustering (MMC, Xing et al., 2003) and Pseudo-metric Online Learning Algorithm (POLA, Shalev-Shwartz et al., 2004) use convex optimization to compute optimal transformations of the original feature space. MMC finds the transformation which maximizes the sum of squared distances between differently labeled inputs, while maintaining an upper bound on the sum of distances between similarly labeled inputs. Weinberger et al. (2006) extend this approach to k-Nearest Neighbor classification. A related approach was taken by Schultz and Joachims (2004) who employ relative comparisons to generate a distance metric. Bar-Hillel et al. (2005) also learn an inner product distance using equivalence constraints. Bilenko et al. (2004) integrate the learning of the distance metric and clustering of the data. The CPCM method shares some similarity in spirit to the supervised methods, but is purely unsupervised; it uses a different optimality criterion. The *blur ratio* could, of course, be optimized by methods similar to those used in the supervised setting.

5 Higher Dimensional Problems

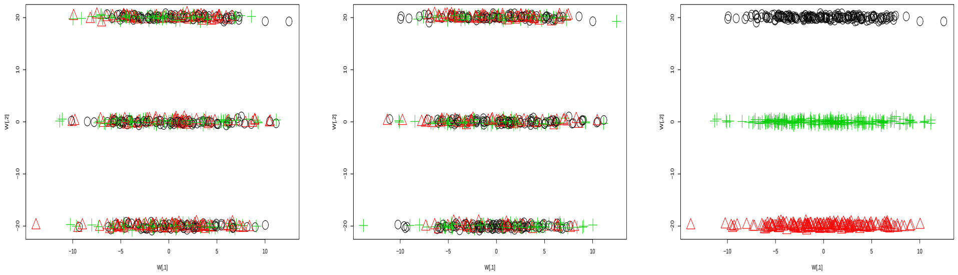
5.1 Simulated datasets

In this example we compare CPCM, LDA-km and Spectral clustering. Cases where these perform better than k-means have already been shown. We take the the three parallel cloud example of Figure 1(d) and add 3 dimensions of high-variance, highly correlated Gaussian noise. Since both LDA-km and CPCM try to cluster in a K-1 dimension, adding these 3 dimensions confuses the algorithms. We now add 50 dimensions of Gaussian noise, which ensures that Spectral clustering fails (though correlated noise suffices too). In our experience, Spectral usually performs poorly in the presence of high-dimensional noise. We run a simple modification of CPCM-LSR where during regression we use a step-wise procedure. This modification underlines the strength of the CPCM framework. By using this, the algorithm is now able to identify the correct clusters. The results when Spectral, LDA-km and CPCM-LSRstep were run are shown in Figure 2. As expected, both spectral [2] and LDA-km produce poor results in this example. CPCM-LSRstep on the other hand identifies the clusters correctly. Without the high-dimensional noise, all three algorithms can identify the clusters correctly.

5.2 UCI machine learning datasets

In this example, we run several different clustering algorithms along with CPCM and LDA-km. We show different criteria to measure and compare the quality of the clusters produced. To compute these criteria we use the given class labels. Table 2(a) shows the *variation of information* criteria of Meila (2005). A lower value is an indicator of better cluster quality. Here CPCM and spectral seem to perform quite well. It is interesting to note that PCA + k-means seems to perform slightly worse than k-means. The rand index was also used for comparison (table 2(b)) and there also CPCM does better than the other algorithms. Classification error using the clusters produced was also calculated and all algorithms perform fairly average, which is not suprising as this is unsupervised learning.

[2] The `kernlab` package (for `R`) implementation of spectral clustering was used.



(a) Spectral (b) LDA-km (c) CPCM-LSRstep

Figure 2: Comparison of Spectral, LDA-km and CPCM on a simulated dataset.

Table 1: Comparison of different algorithms on the UCI dataset

	CPCM - LSR	CPCM - RKHS	Spectral	k-means	k-means + PCA	LDA -km
Glass	0.383	0.371	0.27	0.316	0.37	0.397
Ionosphere	0.205	0.202	0.215	0.198	0.205	0.203
Pima Indian	0.189	0.167	0.136	0.17	0.181	0.2
Sonar	0.236	0.231	0.147	0.257	0.229	0.229
Vehicle	0.323	0.318	0.285	0.325	0.372	0.339
Vowel	0.429	0.465	0.432	0.426	0.466	0.49

(a) Variation of information

	CPCM - LSR	CPCM - RKHS	Spectral	k-means	k-means + PCA	LDA -km
Glass	0.68	0.73	0.627	0.681	0.67	0.665
Ionosphere	0.571	0.501	0.538	0.589	0.573	0.577
Pima Indian	0.516	0.521	0.547	0.551	0.572	0.502
Sonar	0.508	0.499	0.498	0.503	0.499	0.499
Vehicle	0.674	0.568	0.564	0.651	0.648	0.669
Vowel	0.861	0.842	0.709	0.859	0.85	0.852

(b) Rand index

5.3 Gene expressions

Alizadeh et al. (2000) characterize the gene expression of B-cell malignancies which cause diffuse large B-cell lymphoma (DLBCL). They study 96 samples of normal and malignant lymphocytes with 4096 gene expressions each. Using domain expertise, the tissue samples were classified into 9 categories like DLBCL, Germinal centre B and so on. We aim to recreate these $K = 9$ categories using clustering. The distribution of the classes is very uneven $(46, 2, 2, 10, 6, 6, 9, 4, 11)$ making it a tough clustering problem (refer to Figure 1 in (Alizadeh et al., 2000)). We take the first $K - 1$ PCA components as our features and use them for clustering (Ding & He, 2004). For comparisons, k-means and spectral clustering were used in addition to CPCM-RKHS. 100 runs of each algorithm were run and the average Rand indices are reported in table 2. The clustering from CPCM is much better than k-means or spectral in terms of the Rand index.

Table 2: Rand index for different algorithms (averaged over 100 runs) on the Alizadeh et. al. data

	k-means	Spectral	CPCM-RKHS
Rand index	0.748	0.777	0.845

6 Discussion

Learning metrics is critical when features come on very different scales, and when many features are irrelevant or highly correlated. We have shown that CPCM is able to learn transformations of the data that give good clusters in the presence of high dimensional noise, for cases that spectral clustering and LDA-km produce poor results. A big advantage of the proposed method is the flexibility of substituting different prediction algorithms, eg step-wise regression in the example effectively handles correlated noise.

One attractive feature of the transformations that CPCM learns is that it is invariant under non-singular linear transformations of the data. Transforming the original points just results in a compensatory transformation of the distance metric; the clusters remain unchanged.[3] Clearly gaining this invariance must come at a price. Since CPCM considers a much richer space of clusters than standard k-means (or, equivalently, has more free parameters to fit), it could end up fitting more noise than k-means does. Viewed differently, if the clusters are overlapping, it could be the case that k-means does better given the correct metric. Our preliminary empirical studies have shown that in such cases where the linear regression step is not able to find separation, the regression does not warp the data very much, so CPCM returns similar results to k-means. In short, CPCM works well if there are better separated clusters in *some* space generated from a linear combination of the original features.

References

Alizadeh, A. A., et al. (2000). Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature, 403*, 503–511.

Bar-Hillel, A., Hertz, T., Shental, N., & Weinshall, D. (2005). Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, *6*, 937–965.

Bilenko, M., Basu, S., & Mooney, R. J. (2004). Integrating constraints and metric learning in semi-supervised clustering. *ICML '04*.

Chapelle, O., Weston, J., & Schölkopf, B. (2003). Cluster kernels for semi-supervised learning. *NIPS 15*.

Charalampidis, D. (2005). A modified k-means algorithm for circular invariant clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*, 1856–1865.

Dasgupta, S. (1999). Learning mixtures of gaussians. *FOCS '99*.

Ding, C., & He, X. (2004). K-means clustering via principal component analysis. *ICML '04*.

Ding, C., He, X., Zha, H., & Simon, H. (2002). Adaptive dimension reduction for clustering high dimensional data. *ICDM '02*.

Ding, C., & Li, T. (2007). Adaptive dimension reduction using discriminant analysis and k-means clustering. *to be presented in ICML '07*.

Fitzgibbon, A. W., & Zisserman, A. (2002). On affine invariant clustering and automatic cast listing in movies. *ECCV '02*.

Frey, B. J., & Jojic, N. (2003). Transformation-invariant clustering using the em algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*, 1–17.

Friedman, H. P., & Rubin, J. (1967). On some invariant criteria for grouping data. *Journal of American Statistical Association*, 1159–1178.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition*. Academic Press. 2nd edition.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference, and prediction*. New York: Springer.

Hastie, T. J., Buja, A., & Tibshirani, R. (1995). Penalized discriminant analysis. *Annals of Statistics*.

Hastie, T. J., Tibshirani, R., & Buja, A. (1994). Flexible discriminant analysis by optimal scoring. *Journal of American Statistical Association*.

[3] The invariance in this paper is much stronger than *scale invariance* suggested by Kleinberg (2003) in which the distance function is scaled.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys*, *31*, 264–323.

Kannan, R., Vempala, S., & Vetta, A. (2000). On clusterings: Good, bad, and spectral. *FOCS '00*.

Kleinberg, J. (2003). An impossibility theorem for clustering. *NIPS 15*.

Kumar, M., & Orlin, J. B. (2005). Scale-invariant clustering using minimum volume ellipsoids. RUTCOR Research Report.

Meila, M. (2005). Comparing clusterings - an axiomatic view. *ICML '05*.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. *NIPS 14*.

Peng, J., & Xia, Y. (2005). A new theoretical framework for k-means-type clustering. In *Foundations and advances in data mining*, 79–96. Springer-Verlag.

Schultz, M., & Joachims, T. (2004). Learning a distance metric from relative comparisons. *NIPS 16*.

Shalev-Shwartz, S., Singer, Y., & Ng, A. Y. (2004). Online and batch learning of pseudo-metrics. *ICML '04*.

Shental, N., Hertz, T., Weinshall, D., & Pavel, M. (2002). Adjustment learning and relevant component analysis. *ECCV '02*.

Weinberger, K., Blitzer, J., & Saul, L. (2006). Distance metric learning for large margin nearest neighbor classification. *NIPS 18*.

Xing, E. P., Ng, A. Y., Jordan, M. I., & Russell, S. (2003). Distance metric learning with application to clustering with side-information. *NIPS 15*.