A Unified Framework to Integrate Supervision and Metric Learning into Clustering *

Xin Li and Dan Roth
Department of Computer Science
University of Illinois, Urbana, IL 61801
(xli1,danr)@uiuc.edu

December 1, 2004

Abstract

In this paper, we propose a unified framework for applying supervision to discriminative clustering, which: (1) explicitly formalizes the problem of training a partition function as a supervised metric-learning process; (2) learns a partition function that can optimize a supervised clustering error; and (3) flexibly learns a distance metric with regard to any clustering algorithm. Moreover, we develop a general gradient-descent learning algorithm that trains a distance metric under this framework. The convergence of this algorithm is guaranteed for some restricted cases. Our experimental study shows the significant performance improvement after integrating supervision and distance metric learning in clustering, trained in our new framework.

1 Introduction

Many learning tasks aim at partitioning data into disjointed sets (Kamvar, Klein, and Manning, 2002; Bach and Jordan, 2003; Bilenko, Basu, and Mooney, 2004). The hope is to find a good partitioning algorithm that works well to match the standard designed by a human. When labeled data is limited and expensive to obtain, clustering is typically adopted as an unsupervised approach to these tasks. The typical clustering scenario is: given a set of unlabelled data points and a standard distance metric that measures the distance between any two data points, a specific clustering algorithm optimizes a quality function defined over the data set. Clustering quality is typically measured without any supervision. Note that clustering involves no learning this way.

*We would like to thank Vasin Punyakanok and Steve Hanneke for discussions and suggestions on this work.
There are many problems with this framework. First, without any supervision, clustering results could be disparate from a human’s intention. For example, to partition the data points in Figure 1 into $k (k = 4)$ groups, different clustering algorithms with different quality functions and distance metrics can partition them in many different ways, such as horizontally or vertically. Second, this framework lacks flexibility. By using a fixed distance metric and a given clustering algorithm, we enforce a lot of strict assumptions on the data set and the task itself (Kamvar, Klein, and Manning, 2002). For instance, the K-means clustering algorithm works only when the data points are generated from Gaussian distributions with identical variances in the given metric space. Any minor transformation of the metric space and the generative models will lead to clustering failure.

In order to solve these problems, researchers have started to extend clustering in two directions: (1) exploiting supervision to help clustering, and (2) incorporating distance metric learning to increase flexibility. These two directions are closely related in many works. (Bach and Jordan, 2003) attempts to learn a distance metric in an unsupervised setting by optimizing clustering quality. Some works that incorporate partial supervision with distance metric learning include (Bar-Hillel et al., 2003; Schultz and Joachims, 2004; Bilenko, Basu, and Mooney, 2004), but supervision appears only as constraints when they optimize an unsupervised quality function. In these works, supervision is not treated as a direct evaluation and target of clustering, and there is no explicit training stage. (Xing et al., 2002) and (Mochihashi, Kikui, and Kita, 2004) train a distance metric by optimizing a quality over a labeled data set. However, their distance metrics are trained independently of any clustering algorithm. Other works that incorporate supervision into discriminative clustering, include (Bilenko et al., 2003; Cohen, Ravikumar, and Fienberg, 2003), which train a similarity metric through a pairwise classification task — whether a pair of data points are in the same class.

Figure 1: An Example of Unsupervised Clustering. a,b denote the partitioning given by a human and by the K-means clustering algorithm, respectively. Data points of different classes are denoted by different shapes.

In this paper, we develop an integrated framework for supervised discriminative clustering (as shown in Figure 2). In this framework, a clustering task is explicitly split into training and application stages. A partition function, parameterized by a distance metric, is trained in a supervised setting. Distinguishing properties of this framework are: (1) The training stage is formalized as an optimization problem to
learn a partition function, that can minimize a clustering error. (2) The clustering error is well-defined and supervised in the sense that in this evaluation, clustering results are compared with a human’s supervision. (3) The goal of learning and application are inherently integrated; that is, we seek to minimize clustering error both in training and application. (4) The partition function or the distance metric can be learned flexibly with regard to any clustering algorithm. (5) Kernels can be integrated with the distance metric in this model to allow a more complex feature space. Moreover, we develop a general gradient-descent learning algorithm that can train a distance metric under this framework for any clustering algorithm.

In the rest of this paper, we will first introduce our supervised discriminative clustering framework (SDC) and a general learner to train a distance metric in this framework. After that, we present some preliminary experimental results that exhibit the significant performance improvements of our work compared with the K-means clustering algorithm.

2 Supervised Discriminative Clustering (SDC)

2.1 Problem Definition

Partitioning is the task of splitting a set of elements into non-overlapping subsets. Given a set of elements \( S \subseteq X \), a partitioning \( P(S) = \{S_1, S_2, \ldots, S_K\} \) of \( S \) is a disjoint decomposition of \( S \). We associate with a partition \( P \) a partition function \( p \) that maps any set \( S \subseteq X \) to a set of indices \( \{1, 2, \ldots, K\} \). Notice that, unlike a classifier, the image of \( p_S(x) \) of \( x \in S \) under the partition function \( p \) depends on the set \( S \). We will omit the subscript \( S \) in \( p_S(x) \) when clear from the context.

Let \( p \) be the target partition over \( X \), and let \( h \) be any partition function over \( X \). In principle, one can measure the deviation of \( h \) from a given partition \( p \) of \( S \subseteq X \), using a loss function \( l_S(h, p) \rightarrow [0, +\infty) \) defined as \( l_S(h, p) = \frac{1}{|S|} \cdot \{|x \in S|(p_S(x) \neq h_S(x))| \}. \) However, in clustering, we seldom know the mapping from the index of each
Definition 2.1 Supervised Training (SDC): Given an annotated data set \( S \), a set of partition functions \( H \), and the error function \( err_S(h, p)(h \in H) \), the problem is to find an optimal partition function \( h^* \) s.t.

\[
h^* = \arg \min_{h \in H} err_S(h, p).
\]

Definition 2.2 Unsupervised Training: Given an unannotated data set \( S \), a set of possible partition functions \( H \), and a quality function \( q_S(h)(h \in H) \), the problem to find an optimal partition function \( h^* \) s.t.

\[
h^* = \arg \max_{h \in H} q_S(h).
\]

With this formalization, our supervised clustering work can be distinguished clearly from (1) unsupervised clustering approaches - which typically attempt to learn a metric to optimize \( q \) and (2) related works that exploit partial supervision in metric learning – in the training stage of our framework, supervision is directly integrated into the error function and the goal is to find a function in \( H \) that minimizes the clustering error instead of optimizing clustering quality with constraints. Consequently, given new data \( S' \) at the application stage, under some assumptions (not stated in this paper), the hope is that the learned partition function can generalize well and achieve small \( err_{S'}(h, p) \).

2.2 Parameterizing a Partition Function

In reality, a partition function \( h \) in clustering is a pairing of a clustering algorithm \( \mathcal{A} \) and a distance metric \( d \). That is, \( h(S) \equiv \mathcal{A}(d, S) \). Either component could influence the clustering results. Therefore, \( h \) can be parameterized by either \( \mathcal{A} \) or \( d \), or both.
this preliminary work, we only study one of the three options: we fix the clustering algorithm (for example, K-means) and learn a distance metric to minimize the clustering error. The error function \( \text{err}_S(h, p) \), as defined in Equ. 1, is adopted in the rest of this paper.

Suppose each element \( x \in X \) is represented as a feature vector \( x = < f_1, f_2, \ldots, f_m > \) and a distance metric \( d(x, x') \rightarrow [0, +\infty) \) is defined as a function \( d(x, x') = g(< f_1, \ldots, f_m >, < f'_1, \ldots, f'_m >) \). The parameters in \( g \) can be represented as \( \theta \in \Theta \).

Given real-valued features, an example of \( d \) could be

\[
d_1(x, x') = \sum_{k=1}^{m} w_k \cdot |f_k - f'_k|
\]

where \( d_1 \) is parameterized by \( \theta = \{w_k\} \).

**Definition 2.3 Supervised Metric Learning:** Given an annotated data set \( S \), a family of possible distance metrics parameterized by \( \theta \in \Theta \) and a specific clustering algorithm \( A \), and \( \text{err}_S(h, p)(h \in H) \) as defined in Equ. 1, the problem is to seek an optimal partition function \( h^* \) that satisfies

\[
\theta^* = \arg \min_{\theta} \text{err}_S(h, p),
\]

where \( h^*(S) \equiv A(d_{\theta^*}, S) \).

Note that in this framework, a distance metric is learned with respect to a given clustering algorithm. There does not exist a universal distance metric that can satisfy the assumptions of any clustering algorithm. It is necessary to learn a specific distance metric for a given clustering algorithm so that in this metric space, the assumptions of the clustering algorithm could be satisfied.

### 2.3 A General Learner for SDC

In addition to the general framework of supervised discriminative clustering, we describe a general learning algorithm based on gradient descent algorithm that can learn a metric with respect to any clustering algorithm (as shown in Figure 2.3). The gradient descent algorithm converges to a local optima when there exists one. When the error function \( \text{err} \) is linear in \( \theta \), the above algorithm becomes a perceptron-like algorithm: that is, if \( \theta = \{w_k\} \) and \( d(x, x') = \sum_{k} w_k \cdot f_k(x, x') \), where \( f_k(x, x') \) is a feature between \( x, x' \). If we further define a global feature over \( S \) as \( \psi_t(h, S) = \frac{1}{|S|^2} \cdot \sum_{ij} f_t(x_i, x_j) \cdot I[h(x_i) = h(x_j)] \) then the update rule in Step 2(e) becomes

\[
w_k^t = w_k^{t-1} + \psi_k^{t-1}(p, S) - \psi_k^{t-1}(h, S).
\]

The convergence of this algorithm can be induced from Michael Collins’s mistake-bound theorem on the generalized perceptron algorithm (Theorem 1 in (Collins, 2002)): this algorithm converges to the global optima when there exists \( \theta = \{w_k\}(\exists w_k \neq 0) \) that can reach zero error.
Algorithm: SDC-Learner

Input: $S$ and $p$: the annotated data set. $\mathcal{A}$: the clustering algorithm. $err_S(h, p)$: the clustering error function.

Output: $\theta^*$.

1. In the initial (I-) step, we randomly choose $\theta^0$ for $d$. After this step we have the initial $d^0$ and $h^0$.

2. Then we iterate over $t \ (t = 1, 2, \ldots)$,

   (a) Partition $S$ using $h^{t-1}(S) \equiv \mathcal{A}(d^{t-1}, S)$;

   (b) Compute $err_S(h^{t-1}, p)$ and update $\theta$ using the formula:

       \[ \theta^t = \theta^{t-1} + \partial err_S(h^{t-1}, p) / \partial \theta. \]

3. Stopping Criterion: If $err_S(h^t, p) \geq err_S(h^{t-1}, p)$ or $t > T$, the algorithm exits.

Figure 3: A General Training Algorithm for SDC

3 Experimental Study

3.1 Simulations

In this simulation, we randomly generate two data sets. Each data set contains 160 data points that are uniformly generated from four classes on a plane. Each data point has two features $(x, y)$. The K-means algorithm is applied in the training stage of SDC to learn a linear distance metric defined as $d_1$ in Equ. 2 parameterized by $\theta = \{w_x, w_y\}$. The perceptron-like algorithm in Fig. 2.3 with the update rule in Equ. 3 has been applied in training the distance metric with the two labeled data set respectively. In the initialization step, $\theta$ is set to be $(0, 1)$ which focus on the vertical distance between data points. In Figure 4, we can observe that the training effect on the two data sets are significant. Without any supervision, the K-means algorithm will always partition the data points along the rows while true partition is along the columns. After training, the same K-means algorithm with the learned distance metric partitions the data points correctly after 20 iteration steps. The learned metric parameters become $(0.86, 0.14)$ after training for the first data set, focusing on the horizontal distance.

3.2 Real Data Sets

In the following preliminary experiments, we only evaluate the SDC framework on training a good distance metric with respect to the K-means clustering algorithm. The performance is then compared with that of an unsupervised K-means algorithm (with $L_1$ as metric). For this purpose, we collected four data sets from the UCI repository (Blake and Merz, 1998). The K-means algorithm is applied, both in the training stage of SDC to learn a linear distance metric defined as $d_1(\theta = \{w_k\})$ in Equ. 2, and
Figure 4: Two Simple Examples of Training a Linear Distance Metric. Different colors represent different classes. The first sub-figure in each row denotes the true partition of the points in each data set; the second shows unsupervised K-means clustering results with the initial distance metric; and the third sub-figure shows the clustering results using the learned distance metric after training.

The test stage where it, combined with the learned distance metric, is used to cluster test data set. The initial weights are set to 1. The perceptron-like algorithm in Fig. 2.3 with update rule as in Equ. 3 has been applied in training. Moreover, the initial center of each cluster is randomly chosen for the K-means algorithm. Results are averaged on 20 runs of two-folded cross-validation – the unsupervised K-means is only ran on the test sets. The performance of both approaches are evaluated in a standard way – computing precision, recall and $F_1$ over all pairs of elements, relative to the target partition. Not surprisingly, learning a better metric for the data results in that the SDC framework outperforms the unsupervised K-means clustering significantly on these data sets.

<table>
<thead>
<tr>
<th>No.</th>
<th>Data Sets</th>
<th># of elements</th>
<th># of features</th>
<th># of classes</th>
<th>K-Means</th>
<th>SDC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>iris plants</td>
<td>150</td>
<td>4</td>
<td>3</td>
<td>82.59</td>
<td>91.21</td>
</tr>
<tr>
<td>2</td>
<td>breast cancer</td>
<td>569</td>
<td>31</td>
<td>2</td>
<td>78.43</td>
<td>91.64</td>
</tr>
<tr>
<td>3</td>
<td>balance</td>
<td>625</td>
<td>4</td>
<td>3</td>
<td>46.93</td>
<td>52.55</td>
</tr>
<tr>
<td>4</td>
<td>wine</td>
<td>178</td>
<td>13</td>
<td>3</td>
<td>63.26</td>
<td>88.11</td>
</tr>
</tbody>
</table>

Table 1: Comparison of SDC with the unsupervised K-means ($F_1$). Four UCI data sets are used in our experiments. The size of each data set, the # of features and the # of classes are summarized. Results are averaged on 20 runs of two-fold cross validation.
4 Conclusion

In this paper, we propose a unified framework for applying supervision to discriminative clustering, which can be used to learn a distance metric for any given clustering algorithm to minimize a supervised clustering error. Our preliminary experimental study shows its performance improvement and more flexibility over unsupervised clustering.

References


