# Learning a Distance Metric for Object Identification Without Human Supervision

Satoshi Oyama and Katsumi Tanaka

Department of Social Informatics, Graduate School of Informatics,
Kyoto University,
Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, Japan
{oyama, tanaka}@dl.kuis.kyoto-u.ac.jp
http://www.dl.kuis.kyoto-u.ac.jp

**Abstract.** A method is described for learning a distance metric for use in object identification that does not require human supervision. It is based on two assumptions. One is that pairs of different names refer to different objects. The other is that names are arbitrary. These two assumptions justify using pairs of data items for objects with different names as "cannot-be-linked" example pairs for learning a distance metric for use in clustering ambiguous names. The metric learning is formulated using only dissimilar example pairs as a convex quadratic programming problem that can be solved much faster than a semi-definite programming problem, which generally must be solved to learn a distance metric matrix. Experiments on author identification using a bibliographic database showed that the learned metric improves identification F-measure.

## 1 Introduction

Object identification, which is used for example to determine whether the names of people in documents or databases refer to the same person or not, is an important problem in information retrieval and integration. It is most often used for personal name disambiguation, e.g., author identification in bibliographic databases. Citation and bibliographic databases are particularly troublesome because author first names are often abbreviated in citations. Resolving these ambiguities is necessary when evaluating the activity of researchers, but major citation databases such as the ISI Citation Index[1] and Citeseer's Most Cited Authors in Computer Science[2] cannot distinguish authors with the same first name initial and last name.

Object identification problems are generally solved by clustering data containing the target names based on some similarity measure or distance metric [1]. Similarity and distance are important factors in clustering, and an appropriate similarity/distance measure must be used to achieve accurate results. Several methods have been proposed for learning a similarity measure [2,3] or distance

[1] http://isiknowledge.com/
[2] http://citeseer.ist.psu.edu/mostcited.html

J. Fürnkranz, T. Scheffer, and M. Spiliopoulou (Eds.): PKDD 2006, LNAI 4213, pp. 609–616, 2006.

metric [4] from humanly labeled data. One advantage of using a distance metric rather than a general similarity/dissimilarity measure is that it satisfies mathematical properties such as the triangle inequality and can be used in many existing clustering algorithms. One problem in learning a distance metric is that labeling by a person involves costs. In previous research, labeling was given as pairwise feedback, such as two data items are similar and must be in the same cluster ("must-be-linked") or dissimilar and cannot be in the same cluster ("cannot-be-linked"), but disambiguating two people with the same name or similar names is a subtle and time-consuming task even for a person.

We have developed a distance metric learning method that requires no human supervision for object identification. It is based on two assumptions.

**Different names refer to different objects.** In many object identification problems, pairs of different names presumably refer to different objects with few exceptions. For example, two J. Smiths are ambiguous, while J. Smith and D. Johnson cannot be the same person (neglecting, of course, the possibility of false names or nicknames).

**Names are arbitrary.** There is no reason to believe that the data for two people with the same name are more similar than the data for two people with different names. For example, the research papers written by two different J. Smiths are not assumed to be more similar than those written by J. Smith and D. Johnson. We assume that a pair of data items for two people with different names has the same statistical properties as a pair of data items for two people with the same name.

These two assumptions justify the use of pairs of data items collected for different names (for example, J. Smith and D. Johnson) as cannot-be-linked examples for learning a distance metric to be used for clustering data for people with the same or similar names. The learned distance metric that gives good separation of the data for people with different names can be expected to separate the data for different people with the same name as well. These cannot-be-linked example pairs can be formed mechanically without manual labeling. In our setting, no similar (must-be-linked) example pairs are used. After formulating the distance metric learning problem with only dissimilar example pairs as a convex quadratic programming problem, we present experimental results for author identification using a bibliographic database.

## 2 Preliminaries

In this paper, $\boldsymbol{x}^m \in \mathcal{X}$ denotes data (documents or database records) that contain names, where the superscript $m$ is the index for each data item. Each data item $\boldsymbol{x}^m$ is represented as a $D$ dimensional feature vector $(x_1^m, \ldots, x_D^m)^T$, in which each feature corresponds to, for example, a word in a document or an attribute in a database. The superscript $T$ denotes the transpose of a vector or matrix.

Given vector representations of the data, we can define various distance metrics. For the function $d : \mathcal{X} \times \mathcal{X} \to \mathcal{R}$ to be a (pseudo) metric, it must satisfy the following conditions[3]:

$$\begin{aligned} &d(\boldsymbol{x}^m, \boldsymbol{x}^n) \geq 0 \\ &d(\boldsymbol{x}^m, \boldsymbol{x}^n) = d(\boldsymbol{x}^n, \boldsymbol{x}^m) \\ &d(\boldsymbol{x}^m, \boldsymbol{x}^l) + d(\boldsymbol{x}^l, \boldsymbol{x}^n) \geq d(\boldsymbol{x}^m, \boldsymbol{x}^n) \ . \end{aligned}$$

The Euclidean metric treats each feature equally and independently and does not represent interaction among features. Using $D \times D$ matrix $\boldsymbol{A} = \{a_{i,j}\}$, we can define a distance metric in a more general form:

$$\begin{aligned} d_{\boldsymbol{A}}(\boldsymbol{x}^m, \boldsymbol{x}^n) &= \left((\boldsymbol{x}^m - \boldsymbol{x}^n)^T \boldsymbol{A} (\boldsymbol{x}^m - \boldsymbol{x}^n)\right)^{\frac{1}{2}} \\ &= \left( \sum_{i=1}^{D} \sum_{j=1}^{D} a_{i,j} (x_i^m - x_i^n)(x_j^m - x_j^n) \right)^{\frac{1}{2}} \ . \end{aligned}$$

The necessary and sufficient condition for $d_{\boldsymbol{A}}$ being a pseudo metric is that $\boldsymbol{A}$ be a positive semi-definite matrix, in other words, a symmetric matrix in which all eigenvalues are non negative. Xing *et al.* [4] proposed a distance metric learning method in which similar and dissimilar pairs of examples are given, and a matrix $\boldsymbol{A}$ is found that minimizes the sum of the distances between similar pairs while keeping the distances between dissimilar pairs greater than a certain value. However, the optimization problem includes a constraint: matrix $\boldsymbol{A}$ must be positive semi-definite. We thus have a semi-definite programming problem [5], which is harder to solve than a convex quadratic programming problem, like that used in support vector machine learning [6].

## 3 Distance Metric Learning from Only Dissimilar Example Pairs

### 3.1 Problem Formalization

In our setting, only pairs of dissimilar (cannot-be-linked) examples $(\boldsymbol{x}^m, \boldsymbol{x}^n) \in \mathcal{D}$ are given, where $\mathcal{D} \subset \mathcal{X} \times \mathcal{X}$ is the set of paired examples that are considered to be referring to different objects, that is, examples with different names.

We want examples in such a pair to belong to different clusters. To ensure that, we use a matrix $\boldsymbol{A}$ that enlarges the distance between the two examples $d_{\boldsymbol{A}}(\boldsymbol{x}^m, \boldsymbol{x}^n)$. However, multiplying $\boldsymbol{A}$ by a large scalar makes the distance between any two points long and thus not meaningful. Therefore, we introduce a constraint that the norm of matrix $\boldsymbol{A}$ must be a certain constant, say 1, and find the $\boldsymbol{A}$ that induces a long distance between dissimilar examples in a pair while

[3] $d$ becomes a metric in the strict sense when $d(\boldsymbol{x}^m, \boldsymbol{x}^n) = 0$ if and only if $\boldsymbol{x}^m = \boldsymbol{x}^n$.

satisfying the constraint. As the matrix norm, we use the Frobenius norm:

$$\|\boldsymbol{A}\|_F = \left( \sum_{i=1}^{D} \sum_{j=1}^{D} a_{i,j}^2 \right)^{\frac{1}{2}} .$$

We can now formalize distance metric learning from only dissimilar example pairs as an optimization problem:

$$\max_{\boldsymbol{A}} \min_{(\boldsymbol{x}^m, \boldsymbol{x}^n) \in \mathcal{D}} \quad d_{\boldsymbol{A}}(\boldsymbol{x}^m, \boldsymbol{x}^n) \tag{1}$$

$$\text{s.t.} \quad \|\boldsymbol{A}\|_F = 1 \tag{2}$$

$$\boldsymbol{A} \succeq 0 \ . \tag{3}$$

$\boldsymbol{A} \succeq 0$ means that $\boldsymbol{A}$ should be positive semi-definite. Objective function (1) requires finding the $\boldsymbol{A}$ that maximize the distance between the closest example pair. This idea is similar to large margin principles in SVMs [6] and is justified because clustering errors most probably occur at the cannot-be-linked points closest to each other, and keeping these points far from each other reduces the risk of errors.

To simplify the subsequent calculation, we translate the above problem into an equivalent one:

$$\min_{\boldsymbol{A}} \quad \frac{1}{2}\|\boldsymbol{A}\|_F^2 \tag{4}$$

$$\text{s.t.} \quad d_{\boldsymbol{A}}(\boldsymbol{x}^m, \boldsymbol{x}^n) \geq 1 \quad \forall (\boldsymbol{x}^m, \boldsymbol{x}^n) \in \mathcal{D} \tag{5}$$

$$\boldsymbol{A} \succeq 0 \ . \tag{6}$$

### 3.2 Positive Semi-definiteness of Learned Matrix

We now consider an optimization problem consisting of only (4) and (5) without (6). To solve this problem, we introduce the Lagrangean

$$\begin{aligned} L(\boldsymbol{A}, \boldsymbol{\alpha}) =& \frac{1}{2}\|\boldsymbol{A}\|_F^2 + \sum_{(m,n)} \alpha^{(m,n)} \left(1 - d_{\boldsymbol{A}}(\boldsymbol{x}^m, \boldsymbol{x}^n)\right) \\ =& \frac{1}{2}\|\boldsymbol{A}\|_F^2 + \sum_{(m,n)} \alpha^{(m,n)} \left(1 - (\boldsymbol{x}^m - \boldsymbol{x}^n)^T \boldsymbol{A} (\boldsymbol{x}^m - \boldsymbol{x}^n)\right) , \end{aligned} \tag{7}$$

with Lagrange multipliers $\alpha^{(m,n)} \geq 0$.

In the solution of (4) and (5), the derivative of $L(\boldsymbol{A}, \boldsymbol{\alpha})$ with respect to $\boldsymbol{A}$ must vanish; that is, $\frac{\partial L}{\partial \boldsymbol{A}} = 0$. This leads to the following expansion:

$$\boldsymbol{A} = \sum_{(m,n)} \alpha^{(m,n)} (\boldsymbol{x}^m - \boldsymbol{x}^n)(\boldsymbol{x}^m - \boldsymbol{x}^n)^T \ . \tag{8}$$

A necessary and sufficient condition for $D \times D$ matrix $\boldsymbol{A}$ being positive semi-definite is that for all $D$ dimensional vectors $\boldsymbol{v}$, $\boldsymbol{v}^T \boldsymbol{A} \boldsymbol{v} \geq 0$ holds. This is always

the case for a matrix $\boldsymbol{A}$ in the form of (8). Noting that $\alpha^{(m,n)} \geq 0$, we can confirm this as follows:

$$\boldsymbol{v}^T \boldsymbol{A} \boldsymbol{v} = \sum_{(m,n)} \alpha^{(m,n)} ((\boldsymbol{x}^m - \boldsymbol{x}^n)^T \boldsymbol{v})^2 \geq 0 \ .$$

This means that without condition (6), the positive semi-definiteness of $\boldsymbol{A}$ is automatically satisfied. In fact, the optimization problem consisting of only (4) and (5) is a convex quadratic programming problem and can be solved much faster than a semi-definite programming problem with condition (6).

### 3.3 Relationship to Support Vector Machine Learning

Our formalization of learning a distance metric from only dissimilar example pairs is closely related to support vector machine learning. Actually, the optimization problem can be translated into an SVM learning problem [6] and can be solved by existing SVM software with certain settings.

The optimization problem for training an SVM that classifies the data into two classes is as follows [6]:

$$\min_{\boldsymbol{w},b} \ \frac{1}{2} \|\boldsymbol{w}\|^2 \tag{9}$$

$$\text{s.t.} \ \ y^m(\langle \boldsymbol{w}, \boldsymbol{x}^m \rangle + b) \geq 1 \ \ \forall (\boldsymbol{x}^m, y^m) \in \mathcal{T} \ . \tag{10}$$

$\mathcal{T}$ is the set of training examples $(\boldsymbol{x}^m, y^m)$, where $\boldsymbol{x}^m$ is a data vector and $y^m \in \{-1, +1\}$ is the class label. $\langle \boldsymbol{x}, \boldsymbol{z} \rangle$ is the inner product of vectors $\boldsymbol{x}$ and $\boldsymbol{z}$.

Using the Frobenius product

$$\langle \boldsymbol{A}, \boldsymbol{B} \rangle_F = \sum_{i=1}^{D} \sum_{j=1}^{D} a_{i,j} b_{i,j}$$

of two $D \times D$ matrices, we can rewrite the problem of (4) and (5):

$$\min_{\boldsymbol{A}} \ \frac{1}{2} \|\boldsymbol{A}\|_F^2 \tag{11}$$

$$\text{s.t.} \ \ \langle \boldsymbol{A}, (\boldsymbol{x}^m - \boldsymbol{x}^n)(\boldsymbol{x}^m - \boldsymbol{x}^n)^T \rangle_F \geq 1 \ \ \forall (\boldsymbol{x}^m, \boldsymbol{x}^n) \in \mathcal{D} \ . \tag{12}$$

Comparison of (11) and (12) with (9) and (10) reveals that our problem corresponds to unbiased SVM learning ($b = 0$) from only positive data ($y^m = 1$), if we consider the examples and the learned weight of $D \times D$ matrices as $D^2$ dimensional vectors. The expansion form of the SVM solution $\boldsymbol{w} = \sum_m y^m \alpha^m \boldsymbol{x}^m$ makes clear why our method can avoid semi-definite programming. We use only positive examples (cannot-be-linked pairs), thus all the coefficients for the examples become positive in the solution. If we also used negative examples (must-be-linked pairs), the coefficients for these examples become negative and the solution is not always positive semi-definite.

Substituting (8) into (7) gives us the dual form of the problem:

$$\begin{aligned} \max \quad & \sum_{(m,n)} \alpha^{(m,n)} \\ & -\frac{1}{2} \sum_{(m,n)} \sum_{(m',n')} \left( \alpha^{(m,n)} \alpha^{(m',n')} \langle \boldsymbol{x}^m - \boldsymbol{x}^n, \boldsymbol{x}^{m'} - \boldsymbol{x}^{n'} \rangle^2 \right) \\ \text{s.t.} \quad & \alpha^{(m,n)} \geq 0 \ . \end{aligned}$$

These formulas indicate that our learning problem can be solved by using the quadratic polynomial kernel on $D$ dimensional vectors and that we do not need to calculate the Frobenius products between the $D \times D$ matrices. As with standard SVMs, our method can be "kernelized" [7]. By substituting a positive semi-definite kernel function $k(\boldsymbol{x}, \boldsymbol{z}) = \langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{z}) \rangle$ ($\boldsymbol{\phi}(\boldsymbol{x})$ is a map to a higher dimensional space) for the inner product $\langle \boldsymbol{x}, \boldsymbol{z} \rangle$, we can virtually learn the distance metric matrix for a very high (possibly infinite) dimensional feature space by the so-called "kernel trick." In addition, a distance metric for structured data, such as trees or graphs, can be learned with a kernel function defined on the space of such data.

## 4 Experiments

We tested our method on the DBLP data set, which is a bibliography of computer science papers.[4] The entries were made by people, and many author names include the full first name, not only an initial. We assume that the same first and last names refers to the same person.

From among the Most Cited Authors in Computer Science,[5] we selected eight cases of first-initial-plus-surname names, which involve a collapsing of many distinct author names. We retrieved papers written by authors with the same last name and the same first initial from the DBLP data and randomly selected 100 examples for each abbreviated name. Then we abbreviated first names into initials and removed middle names. Training data were built by pairing examples of different abbreviated names, for example, J. Smith and D. Johnson. We used words in titles, journal names, and names of coauthors as features. Since few words appear more than once in a bibliographic entry, we used binary features.

To learn a distance metric, we used SVM$^{light}$[8]. The learned metric was used in clustering the data from the same-first-initial-and-last authors. We used the single-linkage clustering algorithm [9]. The results of clustering were evaluated by referring to the original full names.

The results with the learned metric were compared to the results with two other metrics, one was the Euclidean distance and the other was the IDF weighting [10]. Since each bibliography entry is short and the same word rarely appears

[4] http://dblp.uni-trier.de/

[5] http://citeseer.ist.psu.edu/mostcited.html

**Table 1.** Maximum F-measure values

| Abbreviated name | F-measure | | | Abbreviated name | F-measure | | |
|---|---|---|---|---|---|---|---|
| | Learned | IDF | Euclidean | | Learned | IDF | Euclidean |
| D. Johnson | .644 | .390 | .399 | L. Zhang | .278 | .165 | .158 |
| A. Gupta | .490 | .170 | .169 | H. Zhang | .423 | .226 | .226 |
| J. Smith | .417 | .270 | .292 | R. Jain | .709 | .569 | .552 |
| R. Johnson | .508 | .253 | .227 | J. Mitchell | .640 | .535 | .536 |

more than once in the entry, we did not apply TF weighting. We neither normalized the feature vectors because the lengths of bibliographic entries are rather uniform. The clustering algorithm enables us to specify the number of clusters. We measured the pairwise precision and recall for each number of clusters. The maximum F-measure (harmonic mean of precision and recall [10]) for each combination of name and metric is given in Table 1. Use of the learned metric consistently resulted in the highest F-measure, while the values varied for different names.

## 5 Related and Future Work

Xing *et al.* [4] proposed a distance metric learning from similar and dissimilar example pairs. They formulated the problem as a semi-definite programming problem, and their algorithm needs a full eigenvalue decomposition to ensure that the learned matrix is positive semi-define. Schultz & Joachims [11] proposed a method for learning a distance metric from relative comparison such as "A is closer to B than A is to C." They also formulated the metric learning as a constrained quadratic programming. In their method, the interactions between features are fixed and optimization is applied to a diagonal matrix. Our method can learn a full distance metric matrix by using only cannot-be-linked pairs.

Shalev-Shwartz, Singer & Ng [12] proposed an online learning algorithm for learning a distance metric. Their algorithm does not strictly solve the constrained optimization problem; it finds successive approximate solutions using an iterative procedure that combines a perceptron-like update rule and the Lanczos method to find a negative eigenvalue. While designed for learning from both similar and dissimilar pairs, their algorithm can avoid the eigenvalue problem, as ours does, if it uses only dissimilar example pairs. The performance of the online kernel perceptron algorithm is close to, but not as good as, that of SVMs for the same problem, while saving significantly on computation time [13]. This suggests an interesting direction for future work: adopting online algorithms that learn only from dissimilar examples and comparing the results to those of our learning method.

## 6 Conclusion

We proposed a method for learning a distance metric for use in object identification that is based on two assumptions: different names refer to different objects

and the data for two people with exactly the same name are no more similar than the data for two people with different names. It learns the distance metric from only dissimilar example pairs, which are mechanically collected without human supervision. We formalized our learning problem as a convex quadratic programming problem, which can be efficiently solved by existing SVM software. Experiments using the DBLP data set showed that the learned metric improves F-measure for object identification.

## Acknowledgements

This work was supported in part by a Grant-in-Aid for Scientific Research (No. 16700097) from MEXT of Japan, by a MEXT project titled "Software Technologies for Search and Integration across Heterogeneous-Media Archives," and by a 21st Century COE Program at Kyoto University titled "Informatics Research Center for Development of Knowledge Society Infrastructure."

## References

1. Mann, G.S., Yarowsky, D.: Unsupervised personal name disambiguation. In: Proc. CoNLL-2003. (2003) 33–40
2. Bilenko, M., Mooney, R.J.: Adaptive duplicate detection using learnable string similarity measures. In: Proc. KDD-2003. (2003) 39–48
3. Oyama, S., Manning, C.D.: Using feature conjunctions across examples for learning pairwise classifiers. In: Proc. ECML-2004. (2004) 322–333
4. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.J.: Distance metric learning, with application to clustering with side-information. In: Proc. NIPS-15. (2003) 505–512
5. Vandenberghe, L., Boyd, S.: Semidefinite programming. SIAM Review **38**(1) (1996) 49–95
6. Vapnik, V.N.: Statistical Learning Theory. John Wiley & Sons (1998)
7. Schölkopf, B., Smola, A.: Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond. MIT Press (2002)
8. Joachims, T.: Making large-scale SVM learning practical. In Schölkopf, B., Burges, C., Smola, A., eds.: Advances in Kernel Methods : Support Vector Learning. MIT Press (1999) 169–184
9. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall (1988)
10. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley (1999)
11. Schultz, M., Joachims, T.: Learning a distance metric from relative comparisons. In: Proc. NIPS-16. (2004) 41–48
12. Shalev-Shwartz, S., Singer, Y., Ng, A.Y.: Online and batch learning of pseudo-metrics. In: Proc. ICML-2004. (2004)
13. Freund, Y., Schapire, R.E.: Large margin classification using the perceptron algorithm. Machine Learning **37**(3) (1999) 277–296