Improving Visualization, Scalability and Performance of Multiclass Problems with SVM Manifold Learning

Catarina Silva[1,2] and Bernardete Ribeiro[2]

[1] School of Technology and Management, Polytechnic Institute of Leiria, Portugal
[2] Dep. Informatics Eng., Center Informatics and Systems, Univ. of Coimbra, Portugal
catarina@dei.uc.pt, bribeiro@dei.uc.pt

Abstract. We propose a learning framework to address multiclass challenges, namely visualization, scalability and performance. We focus on supervised problems by presenting an approach that uses prior information about training labels, manifold learning and support vector machines (SVMs).

We employ manifold learning as a feature reduction step, nonlinearly embedding data in a low dimensional space using Isomap (Isometric Mapping), enhancing geometric characteristics and preserving the geodesic distance within the manifold. Structured SVMs are used in a multiclass setting with benefits for final multiclass classification in this reduced space. Results on a text classification toy example and on ISOLET, an isolated letter speech recognition problem, demonstrate the remarkable visualization capabilities of the method for multiclass problems in the severely reduced space, whilst improving SVMs baseline performance.

1 Introduction

Multiclass learning is the problem of assigning labels to instances where the labels are drawn from a finite set of elements and is being increasingly required by modern applications, such as text classification, protein function classification, speech recognition, music categorization and semantic scene classification. The most common approach to such problems is to build upon classification learning algorithms for binary problems, i.e. problems in which the set of possible labels is of size two. Among these algorithms, support vector machines (SVMs) are accepted as one of the best performing methods in many domains [1,2]. When applied to multiclass classification, SVMs are mostly used in their binary version, by reducing a single multiclass problems into multiple binary problems. For instance, a common method is to build a set of binary classifiers where each classifier distinguishes between one of the labels to the rest [3].

The alternative explored in this work is to make use of structured SVMs [4] and cast them to solve multiclass classification problems. The rationale is that having a tool that handles structured outputs, such as graphs or trees, it is possible to build a multiclass classifier [5].

In multiclass classification the challenges are numerous. Feature selection and dimensionality reduction methods must take into account the relevance of features not only to a particular class, as in the binary setting, but to their impact

M. Kolehmainen et al. (Eds.): ICANNGA 2009, LNCS 5495, pp. 370–379, 2009.

on all classes. Initial feature selection and dimensionality reduction are usually carried out in the feature space as a pre-processing step. Several supervised and unsupervised techniques can be applied. Manifold learning strategies, like Isomap (Isometric Mapping) [6], are effective for extracting nonlinear structures from high-dimensional data in pattern recognition [7]. Finding the structure behind the data may be important for a number of reasons, such as data visualization and performance improvement. Graphical depiction of the training and testing sets can potentially be crucial in multiclass applications, since it makes possible to quickly give large amounts of information to a human operator [8]. To this purpose it is appropriately assumed that the data lies on a statistical manifold, or a manifold of probabilistic generative models [9]. It can be regarded as a supervised learning method, where the training labels play a central role. In such a scenario, manifold learning can be used not only with the traditionally associated algorithms, such as K-Nearest Neighbors (K-NN), but also with state-of-the-art kernel-based machines like support vector machines (SVMs) [1].

In this contribution we extend previous work by the authors [10], generalizing its application to multiclass problems. Specifically, we propose the use of manifold learning, with a Isomap based nonlinear algorithm that uses training label information in the dimensionality reduction step, combined with structured multiclass SVMs based on structured SVMs.

The rest of the paper is organized as follows. In the next section, we set the foundations and background for multiclass problems and for the multiclass support vector machines (SVMs) approach. In Section 3, we introduce manifold learning as a supervised dimensionality reduction method. In Section 4, we introduce our approach for the use of manifold learning in multiclass problems, with an Isomap-based nonlinear dimensionality reduction algorithm combined with multiclass SVMs. Experiments and results are described and analyzed in Section 5. Finally, Section 6 addresses conclusions and future work.

2 SVM Multiclass Classification

SVMs are inherently two-class classifiers. The most common technique to implement SVM multiclass classification with $|\mathcal{C}|$ classes in practice has been to build $|\mathcal{C}|$ *one-versus-rest* classifiers (commonly referred to as *one-versus-all*), and to choose the class that classifies the test datum with greatest margin. Another strategy is to build a set of *one-versus-one* classifiers, and to choose the class that is selected by the most classifiers. Although this involves building $|\mathcal{C}|(|\mathcal{C}|-1)/2$ classifiers, the time for training classifiers may actually decrease, because the training data set for each classifier is much smaller.

However, these are not very elegant approaches to solving multiclass problems. A better alternative is provided by the construction of multiclass SVMs, where we build a two-class classifier over a feature vector $\Phi(\mathbf{x}, y)$, derived from the pair consisting of the input features ($\mathbf{x}$) and the class of the datum (y). At test time, the classifier chooses the class

$$y = \arg\max_{y'} \mathbf{w}^T \Phi(\mathbf{x}, y). \tag{1}$$

where $\mathbf{w}$ represents the set of weights that defines the learning machine. The margin during training is the gap between this value for the correct class and for

the nearest other class, and so the quadratic program formulation will require that

$$\forall_i \forall_{y \neq y_i} \mathbf{w}\Phi(\mathbf{x}_i, y_i) - \mathbf{w}\Phi(\mathbf{x}_i, y) \geq 1 - \xi_i, \tag{2}$$

This general method can be extended to give a multiclass formulation of various kinds of linear classifiers. It is also a simple instance of a generalization of classification where the classes are not just a set of independent, categorical labels, but may be arbitrary structured objects with relationships defined between them, usually referred to as structured SVMs.

The algorithm used in this contribution, described in [4], is based on Structured SVMs [5]. It can implement the conventional winner-takes-all (WTA) multiclass classification described in [3]. It learns mappings involving complex structures in polynomial time. A possible application, pertinent to our work is multiclass classification. The multiclass task is tackled by generalizing large margin methods to the broader problem of learning structured responses. The naive approach of treating each structure as a separate class is often intractable, since it leads to a multiclass problem with a very large number of classes. This problem is surpassed specifying discriminant functions that exploit the structure and dependencies within the set of classes $\mathcal{C}$. SVM multiclass uses an algorithm that is different from the one in [3]. It follows the work of Collins [11] on perceptron learning with a similar class of discriminant functions.

Let $\mathcal{C} = \{y_1, \ldots, y_K\}$ be the set of classes and $\mathbf{w} = (\mathbf{v}_1', \ldots, \mathbf{v}_k')'$ be a stack of vectors, where v_k is a weight vector associated with the kth class y_k. Following Crammer and Singer [3], one can then define $F(\mathbf{x}, y_k; \mathbf{w}) =< \mathbf{v}_k, \Phi(\mathbf{x}) >$, where $\Phi(\mathbf{x})$ denotes an arbitrary input representation. These discriminant functions can be equivalently represented by defining a joint feature map as follows $\Psi(\mathbf{x}, \mathbf{y}) \equiv \Phi(\mathbf{x}) \otimes \Lambda^c(\mathbf{y})$. Here Λ^c refers to the orthogonal (binary) encoding of the label y and $\otimes$ is the tensor product which forms all products between coefficients of the two argument vectors.

3 Manifold Learning

Many approaches have been proposed for dimensionality reduction, such as the well-known methods of principal component analysis (PCA) [12], independent component analysis (ICA) [13] and multidimensional scaling (MDS) [14]. All these methods are well understood and efficient and have thus been widely used in visualization and classification. Unfortunately, they share a common inherent limitation: they are all linear methods while the distributions of most real-world multiclass data problems are nonlinear.

An emerging nonlinear dimensionality reduction technique is manifold learning, which is the process of estimating a low-dimensional structure which underlies a collection of high-dimensional data. Manifold learning can be viewed as implicitly inverting a generative model for a given set of observations [15]. Let Y be a d dimensional domain contained in a Euclidean space $\mathbb{R}^d$. Let $f : Y \rightarrow \mathbb{R}^D$ be a smooth embedding for some $D > d$. The goal of manifold learning is to recover Y and f given N points in $\mathbb{R}^D$. Isomap [6] provides an implicit description of the mapping f (or f^{-1}). Given $X = \{\mathbf{x}_i \in \mathbb{R}^D | i = 1 \ldots N\}$ find

$Y = \{\mathbf{y}_i \in \mathbb{R}^d | i = 1 \ldots N\}$ such that $\{\mathbf{x}_i = f(\mathbf{y}_i) | i = 1 \ldots N\}$. Without imposing any restrictions of f, the problem is ill-posed. The simplest case is a linear isometry, i.e. f is a linear mapping from $\mathbb{R}^d \rightarrow \mathbb{R}^D$, where $D > d$.

In Isomap [6] the local neighborhood of each example is preserved, while trying to obtain highly nonlinear embeddings with manifold learning. For data lying on a nonlinear manifold, the *true distance* between two data points is the geodesic distance on the manifold, i.e. the distance along the surface of the manifold, rather than the straight-line Euclidean distance. The main purpose of Isomap is to find the intrinsic geometry of the data, as captured in the geodesic manifold distances between all pairs of data points. The approximation of geodesic distance is divided into two cases. In the case of neighboring points, Euclidean distance in the input space provides a good approximation to geodesic distance. In the case of faraway points, geodesic distance can be approximated by adding up a sequence of *short hops* between neighboring points. Isomap shares some advantages with PCA and MDS, such as computational efficiency and asymptotic convergence guarantees, but with more flexibility to learn a broad class of nonlinear manifolds. The Isomap algorithm takes as input the distances $d(\mathbf{x}_i, \mathbf{x}_j)$ between all pairs $\mathbf{x}_i$ and $\mathbf{x}_j$ from N data points in the high-dimensional input space. The algorithm outputs coordinate vectors $\mathbf{y}_i$ in a d-dimensional Euclidean space that best represent the intrinsic geometry of the data. Isomap is accomplished following these steps:

Step 1. Construct neighborhood graph: Define the graph G over all data points by connecting points $\mathbf{x}_i$ and $\mathbf{x}_j$ if they are closer than a certain distance ε, or if $\mathbf{x}_i$ is one of the K nearest neighbors of $\mathbf{x}_j$. Set edge lengths equal to $d(\mathbf{x}_i, \mathbf{x}_j)$.

Step 2. Compute shortest paths: Initialize $d_G(\mathbf{x}_i, \mathbf{x}_j) = d(\mathbf{x}_i, \mathbf{x}_j)$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are linked by an edge; $d_G(\mathbf{x}_i, \mathbf{x}_j) = +\infty$ otherwise. Then for each value of $k = 1, 2, \ldots, N$ in turn, replace all entries $d_G(\mathbf{x}_i, \mathbf{x}_j)$ by $min\{d_G(\mathbf{x}_i, \mathbf{x}_j), d_G(\mathbf{x}_i, \mathbf{x}_k) + d_G(\mathbf{x}_k, \mathbf{x}_j)\}$. The matrix of final values $\mathbf{D}_G = \{d_G(\mathbf{x}_i, \mathbf{x}_j)\}$ will contain the shortest path distances between all pairs of points in G.

Step 3. Apply MDS to the resulting geodesic distance matrix to find a d-dimensional embedding.

This is an unsupervised procedure and constitutes a preprocessing step for classification. Basically it performs a transformation from a high dimensional input data space into a lower dimensional feature space. Then a classifier, for instance, K-NN can be applied to the resulting data. However, the mapping function given by Isomap is only implicitly defined. Therefore, it should be learned by nonlinear interpolation techniques, such as generalized regression neural networks, which can then transform the new test data into the reduced feature space before prediction.

3.1 Supervised Isomap

In the supervised version of Isomap [16], the information provided by the training class labels is used to guide the procedure of dimensionality reduction. The training labels are used to refine the distances between inputs. The rationale is

that both classification and visualization can benefit when the inter-class dissimilarity is larger than the intra-class dissimilarity. However, this can also make the algorithm overfit the training set and can often make the neighborhood graph of the input data disconnected. To achieve this purpose, the Euclidean distance $d(\mathbf{x}_i, \mathbf{x}_j)$ between two given observations $\mathbf{x}_i$ and $\mathbf{x}_j$, labeled y_i and y_j respectively, is replaced by a dissimilarity measure [16]:

$$D(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} \sqrt{1 - e^{\frac{-d^2(\mathbf{x}_i, \mathbf{x}_j)}{\gamma}}} & y_i = y_j, \\ \sqrt{e^{\frac{-d^2(\mathbf{x}_i, \mathbf{x}_j)}{\gamma}}} - \alpha & y_i \neq y_j. \end{cases} \tag{3}$$

Note that the Euclidean distance $d(\mathbf{x}_i, \mathbf{x}_j)$ is in the exponent and the parameter γ is used to avoid that $D(\mathbf{x}_i, \mathbf{x}_j)$ increases too rapidly when $d(\mathbf{x}_i, \mathbf{x}_j)$ is relatively large. Hence, γ depends on the *density* of the data set and is usually set to the average Euclidean distance between all pairs of data points. On the other hand, α gives a certain possibility to points in different classes to be *closer*, i.e. to be more similar, than those in the same class. This procedure allows a better determination of the relevant features and will definitely improve visualization.

4 Proposed Approach

In this section, we propose a learning framework to address multiclass problems. We propose the combination of manifold learning as a feature reduction step, that increasing scalability, also promotes visualization and performance potentialities.

We start by using manifold learning to construct a reduced representation of the input space. As detailed in Section 3, we use a nonlinear embedding of data in a low dimensional space constructed with the supervised version of Isomap (Isometric Mapping) [16], enhancing geometric characteristics and preserving the geodesic distance within the manifold. Therefore, we use the multiclass training labels in the datasets to provide a better construction of features. We further apply the dissimilarity measure (3) to enhance the baseline Isomap Euclidean distance using label information, with α taking the value of 0.65 and γ the average Euclidean distance between all pairs of training data points.

When a reduced space is reached, our aim is to learn a linear-kernel structured multicass SVM [5] that can be applied in unseen examples. For testing, however, Isomap does not provide an explicit mapping of documents. Therefore we can not generate the test set directly, since we would need to use the labels. Hence, we use a generalized regression neural network (GRNN) [17] with a 0.95 spread to learn the mapping and apply it to each test document, before the SVM prediction phase, as can be gleaned from Figure 1 that summarizes the proposed approach. In the training phase the supervised Isomap procedure, that runs on features and label training instances, is captured by the GRNN using only the features. Furthermore, the reduced featured space ($\mathbb{R}^d$) is the place for the SVM multiclass modeling. When a new testing instance is to be classified, the GRNN maps it from $\mathbb{R}^D$ to $\mathbb{R}^d$ and the SVM multiclass linear-kernel model predicts the class.

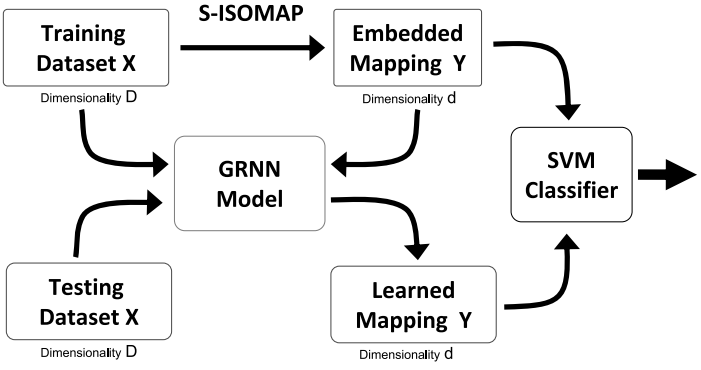


Fig. 1. Proposed approach: SIsomap+SVM

5 Experimental Setup

This section presents the conducted experiments and obtained results. First datasets and performance criteria are defined, then experiments and results are presented and analyzed.

5.1 Datasets

We have used two different multiclass datasets: a text classification toy example[1] and ISOLET, an isolated letter speech recognition problem form UCI[2].

The toy example consists of 7 classes and 2300 examples, divided in 300 training examples and 2,000 testing examples.

ISOLET task is to predict which letter-name was spoken, resulting in a 26-class problem. To generate this dataset, 150 subjects spoke the name of each letter of the alphabet twice. Hence, there are 52 training examples from each speaker. The 617 features are described in [18] and include spectral coefficients, contour features, sonorant features, pre-sonorant features, and post-sonorant features. The examples are split into 6,238 training examples and 1,559 testing examples, both with balanced class cardinality.

5.2 Performance Metrics

In multiclass problems the common performance metric is the global error, given by the percentage of wrongly classified testing instances, regardless of the incorrectly classified category or magnitude of the error.

However, the independent performance of each class is also very important. Therefore, in addition to the global error measure, we also present the error rate and F_1 performances per class. To evaluate each class performance, we first define

[1] http://www.cs.cornell.edu/People/tj/svm_light/svm_multiclass.html
[2] http://archive.ics.uci.edu/ml/datasets/ISOLET

Table 1. Contingency table for binary classification

	Class Positive	Class Negative
Assigned Positive	a	b
	(True Positives)	(False Positives)
Assigned Negative	c	d
	(False Negatives)	(True Negatives)

a contingency matrix representing the possible outcomes of the classification, as shown in Table 1.

Several measures have been defined based on this contingency table, such as, error rate ($\frac{b+c}{a+b+c+d}$), recall ($R = \frac{a}{a+b}$), and precision ($P = \frac{a}{a+c}$), as well as combined measures, such as, the van Rijsbergen F_β measure [19], which combines recall and precision in a single score, $F_\beta = \frac{(\beta^2+1)P \times R}{\beta^2 P+R}$. The F1 measure was chosen since it permits the identification of misclassifications even when a class has few positive examples, detecting deceiving low error rates situations.

5.3 Results and Analysis

Table 2 presents the comparison of global error measures between the baseline multiclass SVM and the proposed approach for both datasets. The feature reduction was from 47 to 10 features for Toy dataset and from 617 to 200 for ISOLET dataset. In the case of ISOLET, to speed the training procedure, of the 6,238 training examples, only 2,500 were used, maintaining balanced class cardinality.

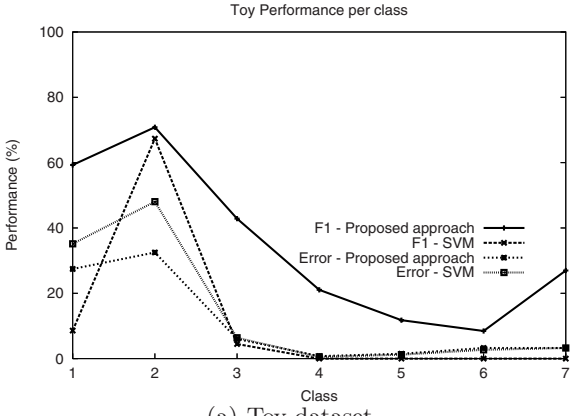
The overall trend is that the proposed manifold multiclass SVM approach surpasses the baseline setting by around 10% and 7% for the Toy dataset and ISOLET dataset respectively (see Table 2).

Figure 2 represents the error rate and F_1 measures for each individual class of the two datasets. The error rates are seamlessly low, while the F_1 performances are more diverse. Nevertheless, the averaged values for error rates for the proposed approach improve the baseline measures: from 13.90% to 10.67% for the Toy dataset and from 1.51% to 0.96% for the ISOLET dataset. Regarding F_1 performance the values vary between the different classes, but the tendency is similar, i.e. the averaged performance values also present an improvement when using the proposed approach: from 11.49% to 34.46% for the Toy dataset and from 79.15% to 87.44% for the ISOLET dataset.

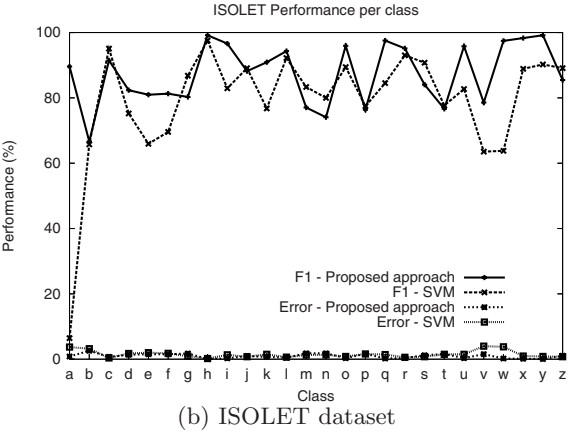
The most impressive result is achieved in visualization properties of the proposed method. As can be gleaned from Figs. 3 and 4, in the initial representation the first ten classes of ISOLET (letters a to j) are not distinguishable, while in

Table 2. Global multiclass error

	SVM	Proposed approach
Toy	48.65%	38.55%
ISOLET	19.63%	12.44%

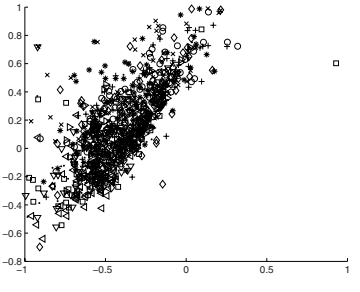


(a) Toy dataset

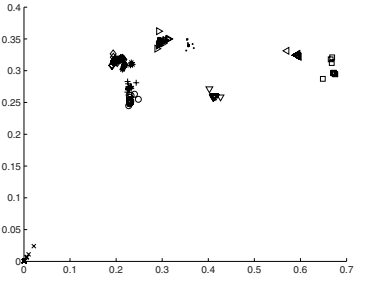


(b) ISOLET dataset

Fig. 2. Performance measures per class for: (a) Toy dataset; (b) ISOLET dataset

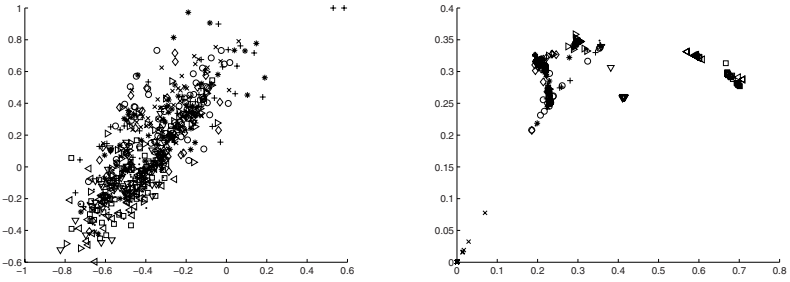


(a) Before manifold learning.



(b) After manifold learning.

Fig. 3. Training examples of ISOLET 10 first letters



(a) Before manifold learning. (b) After manifold learning.

Fig. 4. Testing examples of ISOLET 10 first letters

the manifold representation the ten letters are almost perfectly distinct. Illustration of the 26 classes was attainable with similar results, but the outcome was not graphically clear due to plotting limitations.

6 Conclusions

In this paper we proposed a framework to tackle multiclass problems. We use a combination of a nonlinear dimensionality reduction preprocessing method and structured multiclass SVMs.

We concluded that manifold learning, namely the supervised ISOMAP technique, efficiently captures the underlying structure of the data, preserving the distances among data points in the original dimensional space. One of the main achievements was the impressive graphical class separation that was possible in the manifold. This result can prove to be very useful to transmit information and confidence to a human user. Moreover, the use of structured multiclass SVMs permitted a significant improvement in the performance of the final classifier in the new reduced feature space.

Future work is foreseen in the refinement of the learning abilities and on the exploitation of inter-class relationships in the dimensionality reduction step.

References

1. Vapnik, V.: The Nature of Statistical Learning Theory, 2nd edn. Springer, Heidelberg (1999)
2. Dumais, S., Platt, J., Heckerman, D.: Inductive Learning Algorithms and Representations for Text categorisation. In: ACM Conf. Information Knowledge Management, pp. 148–155 (1998)
3. Crammer, K., Singer, Y.: On the Algorithmic Implementation of Multi-class kernel-based vector machines. Journal Machine Learning Research 2, 265–292 (2002)
4. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: Int. Conf. Machine Learning, pp. 104–111 (2004)

5. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Large Margin Methods for Structured and Interdependent Output Variables. Journal Machine Learning Research 6, 1453–1484 (2005)
6. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 5500(290), 2319–2323 (2000)
7. Kim, H., Park, H., Zha, H.: Distance Preserving Dimension Reduction for Manifold Learning. In: Int. Conf. Data Mining, vol. II, pp. 1147–1151 (2007)
8. Navarro, D., Lee, M.D.: Spatial Visualization of Document Similarity, Defence Human Factors. Special Interest Group Meeting (2001)
9. Zhang, D., Chen, X., Lee, W.: Text Classification with Kernels on the Multinomial Manifold. In: ACM SIGIR - Special Interest Group on Information Retrieval, pp. 266–273 (2005)
10. Silva, C., Ribeiro, B.: Text Classification on Embedded Manifolds. In: Geffner, H., Prada, R., Machado Alexandre, I., David, N. (eds.) IBERAMIA 2008. LNCS (LNAI), vol. 5290, pp. 272–281. Springer, Heidelberg (2008)
11. Collins, M.: Parameter estimation for statistical parsing models: Theory and practice of distribution-free methods. In: IWPT - International Workshop on Parsing Technologies (2001)
12. Jolliffe, I.T.: Principal Component Analysis. Springer, Heidelberg (1986)
13. Comon, P.: Independent Component Analysis: a New Concept? Signal Processing 36(3), 287–314 (1994)
14. Cox, T., Cox, M.: Multidimensional Scaling. Chapman & Hall, London (1994)
15. Duraiswami, R., Raykar, V.C.: The Manifolds of Spatial Hearing. In: ICASSP 2005, vol. III, pp. 285–288 (2005)
16. Geng, X., Zhan, D., Zhou, Z.: Supervised Nonlinear Dimensionality Reduction for Visualization and Classification. IEEE Transactions Systems, Man, and Cybernetics – Part B 35(6), 1098–1107 (2005)
17. Specht, D.: A General Regression Neural Network. IEEE Transactions on Neural Networks 2(6), 568–576 (1991)
18. Fanty, M., Cole, R.: Spoken letter recognition. In: Advances in Neural Information Processing Systems, vol. 3 (1991)
19. van Rijsbergen, C.: Information Retrieval. Butterworths Ed. (1979)