Available online at www.sciencedirect.com
SCIENCE DIRECT
Pattern Recognition 39 (2006) 1839–1851
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

Feature-based approach to semi-supervised similarity learning

Philippe H. Gosselin*, Matthieu Cord

ETIS CNRS UMR 8051, 6 avenue du Ponceau, 95018 Cergy-Pontoise, France

Received 18 April 2006

Abstract

For the management of digital document collections, automatic database analysis still has difficulties to deal with semantic queries and abstract concepts that users are looking for. Whenever interactive learning strategies may improve the results of the search, system performances still depend on the representation of the document collection. We introduce in this paper a weakly supervised optimization of a feature vector set. According to an incomplete set of partial labels, the method improves the representation of the collection, even if the size, the number, and the structure of the concepts are unknown. Experiments have been carried out on synthetic and real data in order to validate our approach.

Keywords: Similarity; Semantic; Concept learning; Statistical; Kernel; Retrieval

1. Introduction

Since the proliferation of digital equipments, the number of digital collections has been growing more and more (public archives, Internet, personal files, etc). In order to manage those large collections, powerful datawarehouse systems are required. Just as librarians classify books according to styles, authors, or dates, one can gather digital documents into *clusters* or *concepts*, for instance, concepts brought by keywords are useful for text database organization. A document can belong to several concepts: for instance, as soon as a science-fiction romance is in the database, *science-fiction* and *romance* keywords will share at least one document. Using concepts, collection management becomes easier, however, those concepts may be unknown *a priori*. A common scheme to discover them is to extract low-level features, for instance, the colors or textures for images, but those features are never perfect, and a learning process is required to retrieve the concepts.

When training samples are available for a concept, a classifier can be built in order to reveal the features [1]. That approach gives excellent results, deals with any combinations of documents, and is able to learn mixed concepts. However, it assumes that the system already knows the concepts likely to be searched.

When one desires a management system for generalist databases, the concepts that users have in mind are unknown beforehand, and sometimes difficult to identify by hand. For instance, concepts for Internet images are unpredictable. However, systems can get advantage of user interaction in order to refine the construction of concepts. Information like "these two documents are similar" or "this document is closer to this one than to that one" can be used in an interactive learning process. Information retrieval techniques like relevance feedback [2–5] and active learning [6,7], which employ user labeling, increase the system performance, but only for the current retrieval session, since the labels are discarded once the session is over.

The user annotations are usually called semantic labels, to differentiate them from the automatically computed document features. This paper deals with concept learning techniques that exploit the semantic labels, for instance, those accumulated during many past uses of a retrieval system. The labels are sampled from a hidden concept that users had in mind during retrieval sessions. The training sets are made

* Corresponding author. Tel.: +33 1 30 73 66 10; fax: +33 1 30 73 66 27.
E-mail address: gosselin@ensea.fr (P.H. Gosselin).

0031-3203/$30.00
doi:10.1016/j.patcog.2006.04.017

of sets of partial labeling corresponding to unknown concepts. Thus, if a large number of labels are available through many retrieval sessions, their combinations can reveal the hidden concepts. We neither make the assumption that the concepts lead to a partition, but are mixed, which means that every document may belong to several concepts. This is a *weakly supervised* learning problem. The aim is to enhance the similarities between documents in order to improve the efficiency of retrieval techniques like clustering, relevance feedback, browsing, etc.

When features match well with concepts, variable selection and feature competition approaches have been proposed to exploit the semantic knowledge [8]. Other methods learn a distance function [9,10]. On the contrary, when concepts are coarsely represented by features, some researchers focus on the similarities between documents. In Ref. [11], an image similarity matrix is updated thanks to the semantic labels. The method enables to modify similarity scores between any couple of images, but has large memory needs. Some researchers propose to perform a clustering of the database in order to reduce the memory needs and to enhance system performances [12]. However, the resulting similarities are difficult to combine with any learning method (classification, active learning, browsing, ...). They usually need a specific learning strategy, keeping out the use of the most powerful ones.

Learning with kernels methods has also been proposed to deal with semantic labels. For two-class problems, a lot of approaches are based on kernel alignment [13]. The idea is to adapt a kernel according to user labeling [14]. As a kernel function can be seen as a particular similarity function, it is possible to improve the similarity between documents. In image retrieval, kernel approaches have been introduced to improve the similarity [15]. In Ref. [16], we recently proposed a kernel matrix updating method, to exploit semantic labels for generalist database management. The method adapts a kernel matrix according to labels provided by the user at the end of each retrieval session. The adaptation aims at reinforcing similarity matrix values for images identically labeled by the user. We introduced some algebraic transformations subject to kernel constraints, in order to always keep the nice properties of kernels. Contrary to the two-class methods, this technique works on generalist image databases, and is designed to model many mixed concepts.

Like Refs. [9,10,16], we propose a learning strategy in a weakly supervised context. However, expressing interesting and efficient data updating rules is not easy when only algebraic transformations subject to constraints are considered. To overcome those difficulties, we propose in this paper a new approach working in the feature space, based on a displacement of feature vectors.

That method arranges feature vectors around a set of equidistant concept centers, without an explicit computation of those centers. According to an incomplete set of partial labels, the method improves the representation of the document collection, even if the size, the number and the structure of the concepts are unknown. Contrary to Refs. [13,14], the method may learn a lot of concepts with many mixed information. Moreover, we address the problem of the complexity in a very efficient way, in opposition to $O(N^2)$ methods like Ref. [10]. The complexity of our technique is no more dependent on the database size, but only on the label set size.

We have built this method in a general framework, thus powerful learning or semi-supervised learning methods may be used to retrieve, classify, or browse data. In particular, we propose to combine the off-line concept learning strategy with SVM classifier and active learning method for online image retrieval.

In this scope, we first present in Section 2 notations, data modeling and problem formulation. In Section 3, we describe the feature-based learning approach. Section 4 presents a semi-supervised context to exploit the concept learning results. Experiments carried out on toy examples and on real image data are reported in Section 5.

2. Challenge

Suppose that we have a set of documents, each of them represented by a vector $\mathbf{x}_i \in \mathbb{R}^{N_d}$ of $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_{N_x}\}$, and a set of labels $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_{N_y}\}$. For instance, X can be the set of feature vectors of an image database, and each $\mathbf{y}_p$ contains the labels provided by a user during the retrieval session p. We denote by $\hat{X}$ the labeled set, the set of vectors with at least one label in Y: $\hat{X} = \{\mathbf{x}_i | \exists p \text{ such as } y_{ip} \neq 0\}$. The couple $(\hat{X}, Y)$ is the training set, and we will talk about *weakly supervised concept learning*, which is the main purpose of this paper. When the unlabeled vectors are also considered, we will talk about *weakly semi-supervised concept learning*.

We suppose that labels are sampled from a hidden set of concepts. The documents are gathered in a finite (but unknown) number N_c of concepts, and those concepts do not necessarily form a partition. Thus, a document represented by a vector $\mathbf{x}_i$ can belong to several concepts. For instance, on an image database, one can find buildings, cars, houses, or landscapes, but also cars in front of a building or a house, or houses in a landscape.

A vector $\mathbf{y}_p \in [-1, 1]^{N_x}$ is a partial labeling of the set X, according to one of the concepts. Every positive value y_{ip} means that the document represented by $\mathbf{x}_i$ is in that concept, as much as y_{ip} is close to 1. Every negative value y_{ip} means that the document represented by $\mathbf{x}_i$ is not in that concept, as much as y_{ip} is close to -1. Every value y_{ip} close to zero means that there is no information for $\mathbf{x}_i$ about that concept. We also suppose that the number of non-zero values in $\mathbf{y}_p$ is small against the size of the concept. Thus, from only one $\mathbf{y}_p$, it is impossible to build the corresponding concept.

The challenge is to use this set of partial labeling in order to learn the concepts.

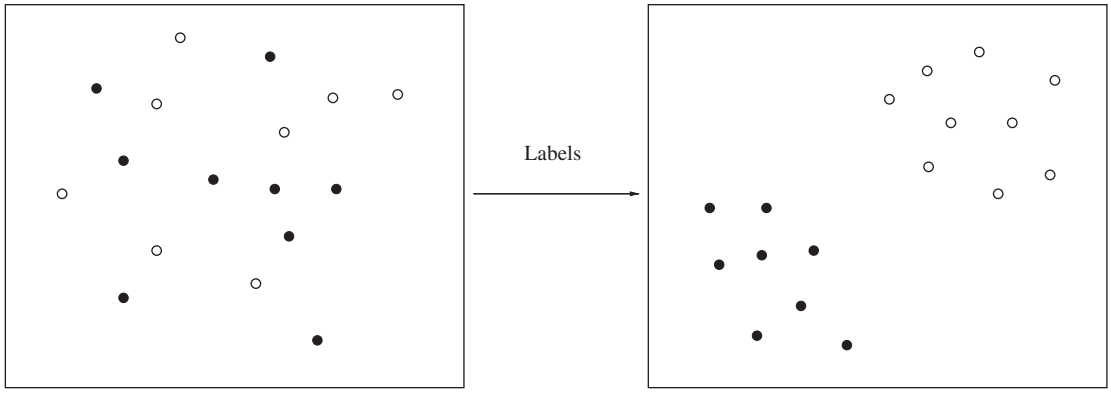

Fig. 1. Concept vector learning principle: vectors are moved according to labels, in order to build clusters.

2.1. Basic concept vector updating method

If we assume that the N_c concepts are represented by a set of centers $C = \{\mathbf{c}_1, \ldots, \mathbf{c}_{N_c}\}$, $\mathbf{c}_j \in \mathbb{R}^{N_d}$, then the membership of a document $\mathbf{x}_i$ to a concept $\mathbf{c}_j$ can be its Euclidean distance to $\mathbf{c}_j$. For instance, if a vector $\mathbf{x}$ is close to $\mathbf{c}_1$ and $\mathbf{c}_2$, then it belongs to the concepts 1 and 2, and does not belong to any other. However, these centers are not available without learning. For a given labeling $\mathbf{y}$, we propose to move each labeled $\mathbf{x}_i$ according to a estimated concept center $\hat{\mathbf{c}}_i$ of the true concept center $\mathbf{c}_i$.

This problem can be seen as a "turned over" K-means problem. During a K-means clustering, vectors are already in clusters, and the algorithm move centers in order to find them. In our context, vectors are not in clusters, and the algorithm moves vectors in order to build clusters (cf. Fig. 1).

A basic approach is to use the barycenter of positive labeled vectors as an estimation of $\mathbf{c}_i^+$:

$$\hat{\mathbf{c}}_i^+ = \mathbf{g}^+ = \frac{1}{\sum_{j\in I^+} y_j} \sum_{j\in I^+} y_j \mathbf{x}_j \tag{1}$$

and the barycenter of negative labeled vectors as an estimation of $\mathbf{c}_i^-$:

$$\hat{\mathbf{c}}_i^- = \mathbf{g}^- = \frac{1}{\sum_{j\in I^-} y_j} \sum_{j\in I^-} y_j \mathbf{x}_j \tag{2}$$

with $I^+ = \{j | y_j > 0\}$ and $I^- = \{j | y_j < 0\}$.

Next, each labeled vector is moved towards its corresponding center (cf. Fig. 2):

$$\mathbf{x}_i \text{ moves towards } \begin{cases} \hat{\mathbf{c}}_i^+ & \text{if } y_i > 0, \\ \mathbf{x}_i & \text{if } y_i == 0, \\ \hat{\mathbf{c}}_i^- & \text{if } y_i < 0. \end{cases}$$

2.2. Global scheme

In order to process the whole set of labels Y, we can perform several updates by randomly picking a label $\mathbf{y}(t) = \mathbf{y}_{rand()\%N_y}$:

$$X(t+1) = \text{update}(X(t), \mathbf{y}(t), \rho(t)).$$

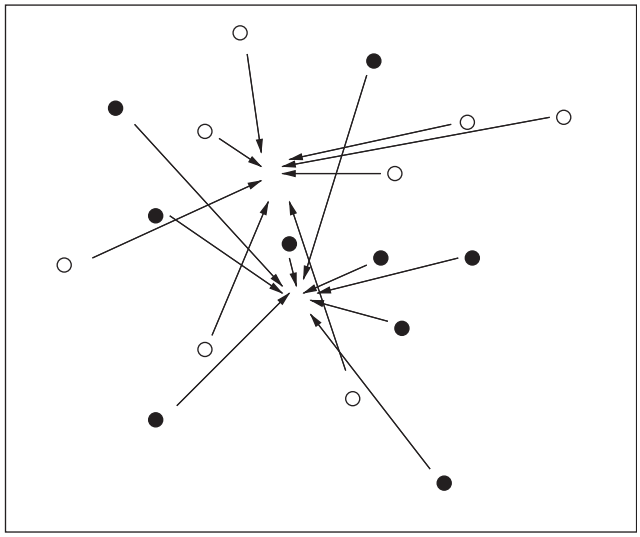

Fig. 2. Basic learning strategy: vectors are moved towards the barycenters.

For each update t, we compute a possible concept center $\hat{\mathbf{c}}_i(t)$ for each vector $\mathbf{x}_i(t)$. Next, we move the labeled vectors towards their corresponding centers:

$$\forall i \in 1 \ldots N_x,\ \mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \rho(t)|y_i(t)|(\hat{\mathbf{c}}_i(t) - \mathbf{x}_i(t)).$$

Repeating this update several times with a decreasing $\rho(t)$, the set $X(t)$ converges to a set $X^\star$ as t tends to the infinity.

Using this global scheme, with a basic update like the one described in the previous section, is efficient only in very specific cases. When vectors are already somewhat gathered into clusters, this basic method may improve the representation. In other cases, for instance the ones in Figs. 1 or 2, vectors are gathered into close clusters. After several updates, all vectors quickly move to the same area,

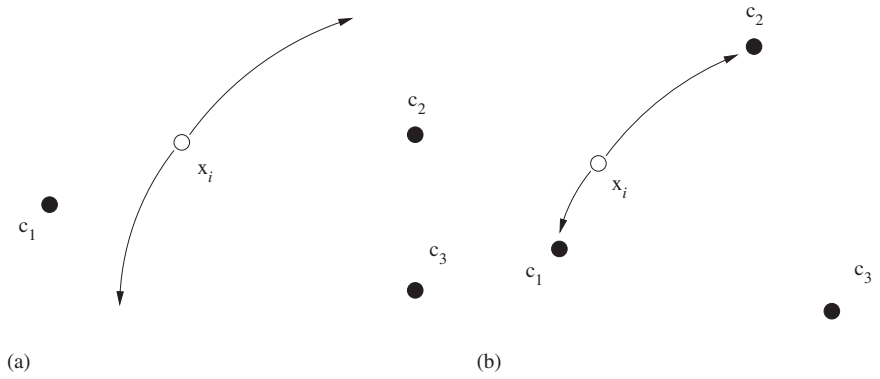

Fig. 3. Non-equidistant (a) and equidistant (b) concept centers.

and concepts are not well separated. In the following section, we propose a method with the same global scheme, but with a more efficient estimation of the centers.

3. Concept vector learning method

We propose in this section a method for learning a representation of data from a training set $(\hat{X}, Y)$, compound of a set of vector $\hat{X} = \{\mathbf{x}_i \in X | \exists p \text{ such as } y_{ip} \neq 0\}$ and a set of labels $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_{N_y}\}$. The method arranges vectors in $\hat{X}$ around a set of concept centers $\mathbf{c}_j$, without explicit computing of the $\mathbf{c}_j$.

3.1. Equidistance constraint

The repartition of the centers in the space is important in the case of mixed concepts. As we wish to represent any possible combination of memberships, centers should be at the same distance from each other.

For instance, let be two close concept centers $\mathbf{c}_2$ and $\mathbf{c}_3$, a third one $\mathbf{c}_1$ far from $\mathbf{c}_2$ and $\mathbf{c}_3$, and a vector $\mathbf{x}$ (cf. Fig. 3). Then moving $\mathbf{x}$ towards $\mathbf{c}_2$ will necessarily move it close to $\mathbf{c}_3$. Thus, it is impossible, for instance, to get $\mathbf{x}$ close to $\mathbf{c}_1$ and $\mathbf{c}_2$, and far from $\mathbf{c}_3$. However, in the case of equidistant centers, this is always feasible.

The building of equidistant centers has implications on the dimension N_d of vectors. Theorem 1 shows that, if all centers are normalized and equidistant, the dimension N_d must be superior or equal to the number of concepts N_c, and in a special case to $N_c - 1$.

Theorem 1. *Let* $C = \{\mathbf{c}_1, \ldots, \mathbf{c}_{N_c}\}$ *be a set of vectors* $\mathbf{c}_j \in \mathbb{R}^{N_d}$. *If* $\forall j = 1 \ldots N_c,\ \|\mathbf{c}_j\| = 1$, *and* $\exists d > 0\ \forall j,\ j' = 1 \ldots N_c,\ \|\mathbf{c}_j - \mathbf{c}_{j'}\|^2 = d$, *then*

$$\text{If } d = 2\left(1 + \frac{1}{N_c - 1}\right) \quad \text{then } N_d \geqslant N_c - 1$$
$$\text{otherwise } N_d \geqslant N_c.$$

In other words, in order to build N_c concepts around equidistant centers, the dimension of vectors must be at least $N_c - 1$.

The theorem also shows that, for N_c concepts in a space of $N_c - 1$ dimensions, that distance between equidistant concept centers is unique, modulo a scaling. It is easy to see that in higher dimension, this property is no longer true. In the computation of possible centers, described in the following, we exploit this property in order to get equidistant centers.

3.2. Positive center computing

Positive labels in $\mathbf{y}$ mean that the corresponding vectors are in the same concept. We could choose to compute the positive barycenter as in Eq. (1). However, this approach does not dispatch vectors into the space, and the problem evoked in the previous section will appear.

The area where the positive labeled vectors are moved is not the most important—what matters the most is the fact that they are gathered, and that the negative labeled vectors are not in the same area. In the case where the positive and negative labeled vectors are not in the same area, positive labeled vectors only need to be gathered in the area they already are. In the case where the positive and negative labeled vectors are in the same area, positive labeled vector should be gathered in a different area.

In order to get this behavior, we propose to compute the barycenter $\mathbf{g}$ of all labeled vectors:

$$\mathbf{g} = \frac{1}{\sum_j |y_j|} \sum_j y_j \mathbf{x}_j.$$

And next, to compute its normalization to check Theorem 1 conditions:

$$\hat{\mathbf{c}}^+ = \frac{\mathbf{g}}{\|\mathbf{g}\|}.$$

We can see, as $\mathbf{g} = \mathbf{g}^+ - \mathbf{g}^-$, that in the case where the positive and negative labeled vectors are not in the same area, for instance in an extreme case where $\mathbf{g}^- = -\mathbf{g}^+$, then $\mathbf{g} = 2\mathbf{g}^+$, and positive labeled vectors gather in the same area. In the

case where the positive and negative labeled vectors are in the same area, for instance in an extreme case where $\mathbf{g}^- \simeq \mathbf{g}^+$, then $\mathbf{g} \simeq 0$, and as $\mathbf{g}$ is normalized, then the positive labeled vectors are gathered in a random area.

Another motivation for this choice of positive center computing is based on the kernel alignment [13]. Seen as a two-concept problem, the problem can expressed as follows:

$$\begin{cases} \min_{X^+} \sum_{i \in I^+} \|\mathbf{x}_i - \mathbf{g}\|, \\ \max_{X^-} \sum_{i \in I^-} \|\mathbf{x}_i - \mathbf{g}\| \end{cases} \tag{3}$$

with $X^+ = \{\mathbf{x}_i | y_i > 0\}$ and $X^- = \{\mathbf{x}_i | y_i < 0\}$.

If we assume that all vectors are normalized, then Eq. (3) is equivalent to

$$\begin{cases} \max_{X^+} \sum_{i \in I^+} \langle \mathbf{x}_i, \mathbf{g} \rangle, \\ \min_{X^-} \sum_{i \in I^-} \langle \mathbf{x}_i, \mathbf{g} \rangle. \end{cases} \tag{4}$$

As $\mathbf{g} = (1/\sum_j |y_j|) \sum_j y_j \mathbf{x}_j$, then Eq. (4) is equivalent to

$$\begin{aligned} &\max_X \sum_i y_i \langle \mathbf{x}_i, \mathbf{g} \rangle \\ &\quad \Longleftrightarrow \max_X \sum_i y_i \left\langle \mathbf{x}_i, \sum_j y_j \mathbf{x}_j \right\rangle \\ &\quad \Longleftrightarrow \max_X \sum_{i,j} y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ &\quad \Longleftrightarrow \max_X A_X(\mathbf{y}), \end{aligned} \tag{5}$$

where $A_X(\mathbf{y})$ is the alignment of the linear kernel $X^t X$ and $\mathbf{y}$.

This shows that the proposed choice for positive center computing is correlated with the kernel alignment. However, this kernel alignment-based approach (cf. Eq. (3)) deals with two-concept problems: it gathers negative labeled vectors, whereas they are not necessarily in the same concept.

We propose in the next subsection a negative center computing, based on equidistance of concept centers, which deals with more than two concepts.

3.3. Negative center computing

Negative labels in $\mathbf{y}$ mean that the corresponding vectors are not in the concept. We proposed to the negative labeled vectors such as the concept centers are equidistants.

Theorem 1 shows that the squared distance between centers should be $d = 2(1 + 1/(N_c - 1))$, in order to have equidistant centers in the smallest space. We choose as an estimated center $\hat{\mathbf{c}}_i^-$ the vector in the plan spanned by $\mathbf{x}_i$ and $\hat{\mathbf{c}}^+$ such as its distance to $\hat{\mathbf{c}}^+$ is $\sqrt{d}$ (cf. Fig. 4). Note that N_c is unknown, but as we will see in experiment, if we take a too large value for $N_d \geqslant N_c - 1$, the method converges as well as if we choose the best one.

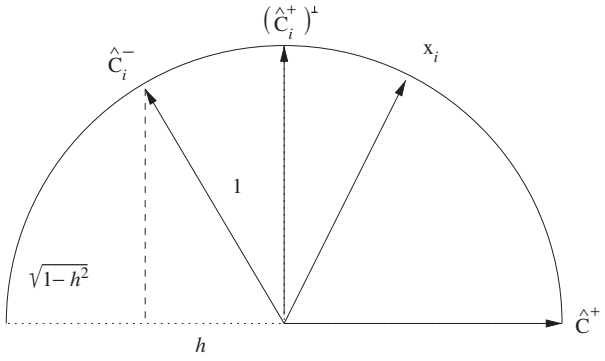

Fig. 4. Possible center $\hat{\mathbf{c}}_i^-$ for a negative labeled vector $\mathbf{x}_i$, relatively to the estimated concept center $\hat{\mathbf{c}}^+$.

for $t = 1 : T$
  $\mathbf{y} = Y_{rand()\%N_y}$
  $\hat{\mathbf{c}}^+ = \frac{\sum_j y_j \mathbf{x}_j}{\|\sum_j y_j \mathbf{x}_j\|}$
  for $i \in I^+$
    $\mathbf{x}_i = \mathbf{x}_i + \rho |y_i| (\hat{\mathbf{c}}^+ - \mathbf{x}_i)$
  endfor
  for $i \in I^-$
    $r = < \mathbf{x}_i, \hat{\mathbf{c}}^+ >$
    $s = \sqrt{\frac{1-h^2}{<\mathbf{x}_i, \mathbf{x}_i>^2 - r^2}}$
    $\mathbf{x}_i = \mathbf{x}_i + \rho |y_i| \big((h - rs)\hat{\mathbf{c}}^+ + (s-1)\mathbf{x}_i \big)$
  endfor
endfor

Fig. 5. Concept vector learning algorithm.

We first compute a basis $\{\hat{\mathbf{c}}^+, (\hat{\mathbf{c}}_i^+)^\perp\}$ of this plan:

$$\begin{aligned} (\hat{\mathbf{c}}_i^+)^\perp &= \frac{\mathbf{x}_i - \langle \mathbf{x}_i, \hat{\mathbf{c}}^+ \rangle \hat{\mathbf{c}}^+}{\|\mathbf{x}_i - \langle \mathbf{x}_i, \hat{\mathbf{c}}^+ \rangle \hat{\mathbf{c}}^+\|} \\ &= \frac{\mathbf{x}_i - \langle \mathbf{x}_i, \hat{\mathbf{c}}^+ \rangle \hat{\mathbf{c}}^+}{\sqrt{\langle \mathbf{x}_i, \mathbf{x}_i \rangle^2 - \langle \mathbf{x}_i, \hat{\mathbf{c}}^+ \rangle^2}}. \end{aligned}$$

Next, as we need to have $\|\hat{\mathbf{c}}^+ - \hat{\mathbf{c}}_i^-\|^2 = d$, then $\langle \hat{\mathbf{c}}^+, \hat{\mathbf{c}}_i^- \rangle = h = -1/N_d$, and:

$$\begin{aligned} \hat{\mathbf{c}}_i^- &= h\hat{\mathbf{c}}^+ + \sqrt{1-h^2}(\hat{\mathbf{c}}_i^+)^\perp \\ &= h\hat{\mathbf{c}}^+ + \sqrt{\frac{1-h^2}{\langle \mathbf{x}_i, \mathbf{x}_i \rangle^2 - \langle \mathbf{x}_i, \hat{\mathbf{c}}^+ \rangle^2}} (\mathbf{x}_i - \langle \mathbf{x}_i, \hat{\mathbf{c}}^+ \rangle \hat{\mathbf{c}}^+) \\ &= \left(h - \langle \mathbf{x}_i, \hat{\mathbf{c}}^+ \rangle \sqrt{\frac{1-h^2}{\langle \mathbf{x}_i, \mathbf{x}_i \rangle^2 - \langle \mathbf{x}_i, \hat{\mathbf{c}}^+ \rangle^2}} \right) \hat{\mathbf{c}}^+ \\ &\quad + \sqrt{\frac{1-h^2}{\langle \mathbf{x}_i, \mathbf{x}_i \rangle^2 - \langle \mathbf{x}_i, \hat{\mathbf{c}}^+ \rangle^2}} \mathbf{x}_i. \end{aligned}$$

The complete method is summarized in Fig. 5. Its complexity depends on the dimension N_d of vectors, the number

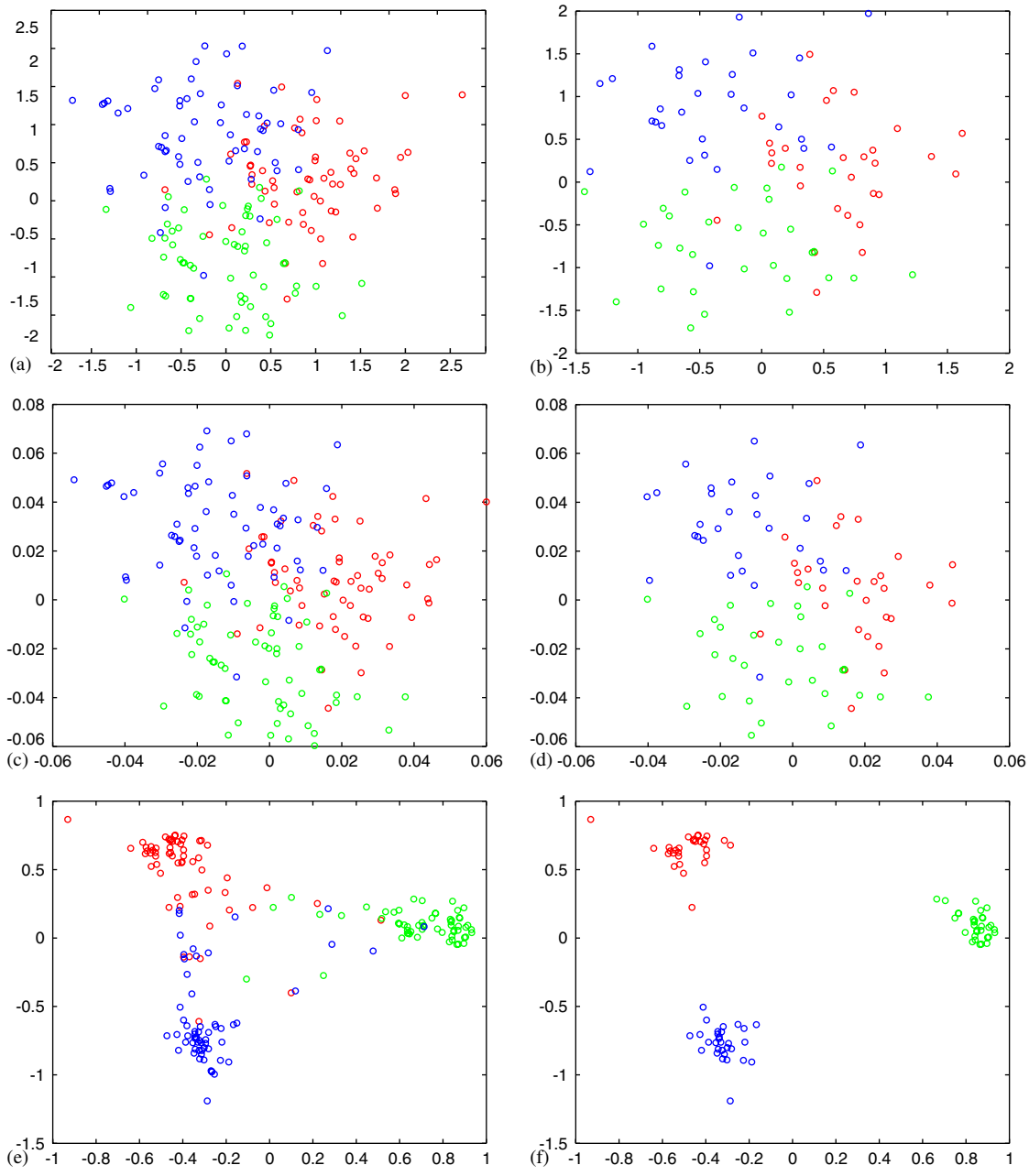

Fig. 6. Toy example with three concepts. Legend: blue = concept 1, red = concept 2, green = concept 3. (a) Initial set; (b) labeled set; (c) optimized set with Xing method; (d) optimized labeled set with Xing method; (e) optimized set with our method; (f) optimized labeled set with our method.

N_l of non-zero values in labels $\mathbf{y}_p$, and the number T of updates. The computation is very fast, and complexity does not depends on the size of the database. In the following experiments on real data the computation takes 1 s with a Pentium 3 Ghz, $N_d = 50$, $N_l = 50$, $T = N_y/\rho$, $N_y = 1000$, and $\rho = 0.1$.

4. Semi-supervised learning

A set of labels Y does not necessarily give knowledge about every vector in X. In order to process these unlabeled vectors, we propose to compute their position according to their similarity in the initial state.

At the end of learning, the set X converges to the set $X^\star$ according to Y. Let $\bar{I} = \{i | \forall p = 1 \dots N_y,\ y_{ip} = 0\}$ be the indexes of unlabeled vectors. At this state, all unlabeled vectors in $X^\star$ are the same than in X. We propose to compute a new position for these vectors using a kernel function $k(.,.)$:

$$\forall i' \in \bar{I}\, \mathbf{x}^\star_{i'} = \frac{\sum_i k(\mathbf{x}_i, \mathbf{x}_{i'}) \mathbf{x}^\star_i}{\sum k(\mathbf{x}_i, \mathbf{x}_{i'})}.$$

Using a gaussian kernel, this approach is a smooth version of a k-nearest neighbors. The width of the gaussian determines the amount of generalization, depending on the quality of

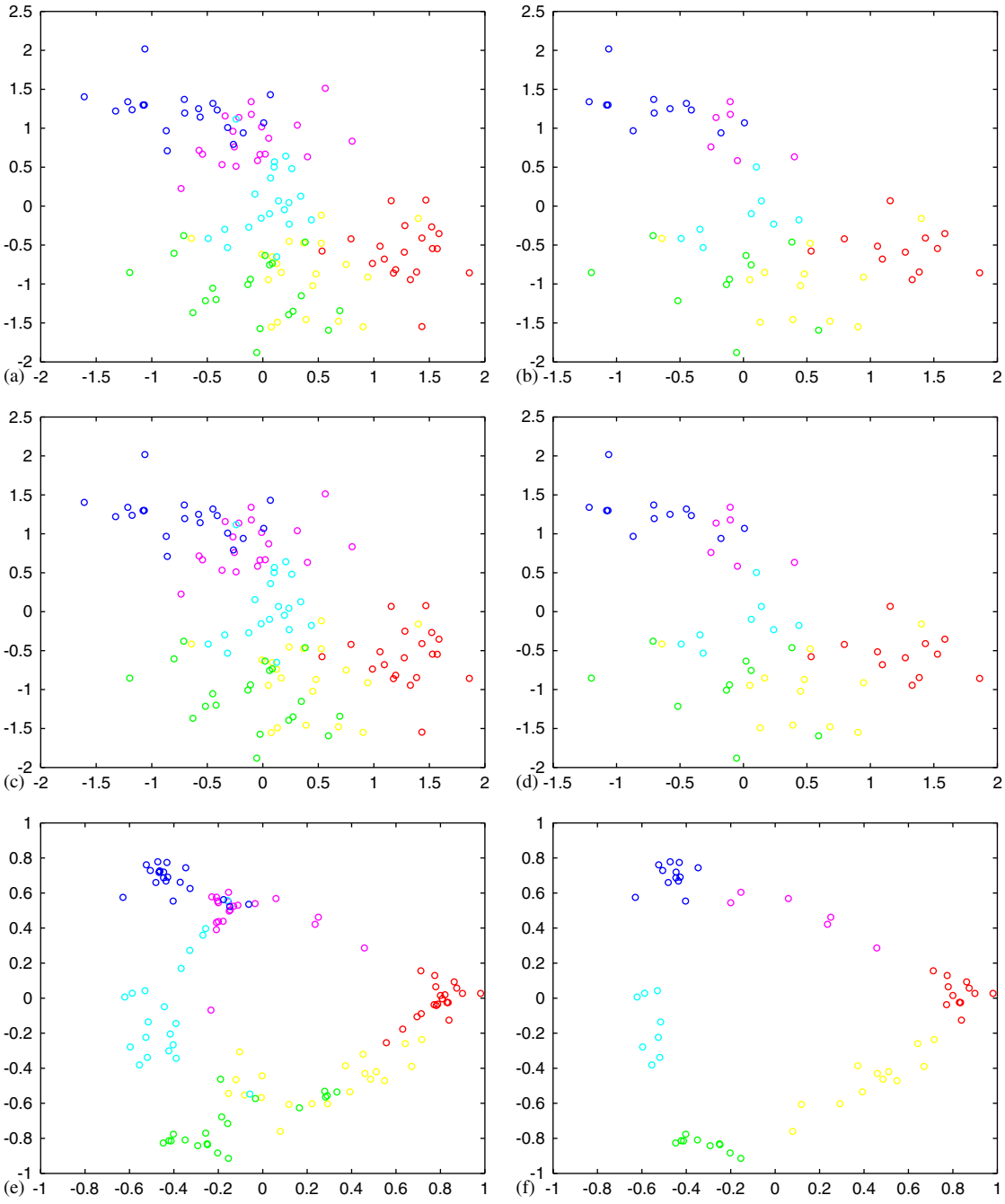

Fig. 7. Toy example with three mixed concepts. Legend: blue=concept 1, red=concept 2, green=concept 3, magenta=concepts 1 and 2, cyan=concepts 1 and 3, yellow = concepts 2 and 3. (a) Initial set; (b) labeled set; (c) optimized set with Xing method; (d) optimized labeled set with Xing method; (e) optimized set with our method; (f) optimized labeled set with our method.

the initial distances one can make on a set X. For small gaussians, few vectors are close to a given vector. Thus, the vector is close to one of the concept centers. In this case, the vector is in one of the concepts, reflecting the confidence in information. For large gaussians, a lot of vectors are close to a given vector. Thus, the vector is close to the center of gravity of the whole set. In this case, the distance of the vector is approximatively the same to all other ones, reflecting the lack of information.

5. Experiments

Notations:

- *The initial set* X . This is the set of the N_x vectors of $\mathbb{R}^{N_d}$ before any learning. These vectors can be automatically computed using color, texture, keywords, etc. Note that the method does not require a good match of these vectors with the concepts, and is able to learn

from randomly sampled data, if the training set is large enough.

- *The partial labeling* Y. This is the set of the N_y partial labels $\mathbf{y}_p$ of $[-1, 1]^{N_x}$. Note that in the following experiments we use integer labels, i.e. $\mathbf{y}_p \in \{-1, 0, 1\}^{N_x}$. These vectors $\mathbf{y}_p$ are the labels provided by users on a retrieval system. We will refer to the number $N_l \ll N_x$ of non-zero values in each of them. This number N_l is insufficient to rebuild a concept using a single $\mathbf{y}_p$.
- *The labeled set* $\hat{X}$. This is the set of vectors with at least one label in matrix Y: $\hat{X} = \{\mathbf{x}_i | \exists p \text{ such as } y_{ip} \neq 0\}$.
- *The training set* $(\hat{X}, Y)$. This set is used for the concept learning.
- *The optimized set* $X^\star$. This is the output of the concept learning method.
- *The optimized labeled set* $\hat{X}^\star$. This set is composed of the vectors of $X^\star$ with at least one label in Y.

5.1. Toy examples with $N_d = N_c - 1$

We build two toy examples with $N_x = 120$ vectors of dimension $N_d = 2$: one with $N_c = 3$ concepts (Fig. 6a) and another one with $N_c = 3$ mixed concepts (Fig. 7a). "Mixed concepts" means that some of the vectors are in two concepts. For instance, magenta vectors in Fig. 7a are in concept 1 (blue + magenta + cyan) and concept 2 (red + magenta + yellow). In both cases, we build a set of labels Y compound of $N_y = 100$ labels $\mathbf{y}_p$ with $N_l = 10$ non-zero values. The labels are not necessarily balanced, a vector $\mathbf{y}_p$ can have more positive than negative values, and vice versa. All vectors with a non-zero value in Y form the labeled set $\hat{X}$ (Figs. 6b and 7b). In these toy examples, the labeled set is compound of 60 vectors: half of X is unlabeled.

Our approach is compared to the distance learning method proposed by Xing [10]. In their technique, the similarity matrix S is $S = (YY' > 0)$ and the dissimilarity matrix is $D = (YY' < 0)$. As one can see in Figs. 6c, d, 7c and d, their distance learning method does not change a lot the configuration of the data.

With our concept vector learning method, the vectors gather into clusters (cf. Figs. 6f and 7f). In the mixed concept case (Fig. 7f), the vectors sharing two concepts are distributed between both concepts, and stay as far as possible from the concept they do not have. The unlabeled vectors are well distributed into the space using the method in Section 4 (Figs. 6e and 7e). However, some vectors are not distributed correctly. For instance, in Fig. 6e, a blue vector appears in the green cluster. This blue vector correspond to a blue vector surrounded by green vectors in the left of Fig. 6a. We can see that this blue vector is not in the labeled set (Fig. 6b). In such a particular case, it is very difficult to find the true concept of a vector, and this is the reason why some unlabeled vectors are not distributed correctly, whereas most of them are well distributed.

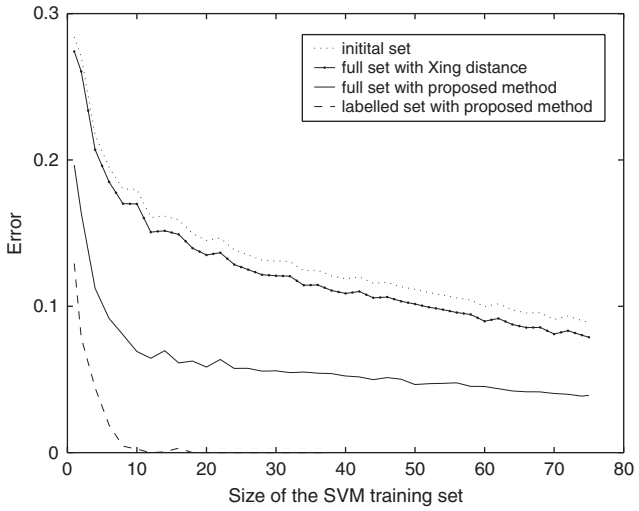

Fig. 8. Toy example with three concepts. SVM-classification errors according to the size of the training set.

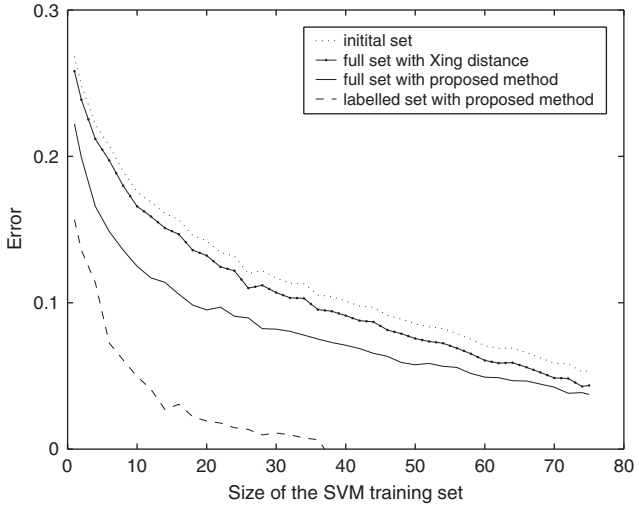

Fig. 9. Toy example with three mixed concepts. SVM-classification errors according to the size of the training set.

One can see that classical learning methods will be much more efficient with such a distribution of data. For instance, a K-means clustering is easier in the case of Fig. 6f than in the case of Fig. 6b. An hyperplane classification (Perceptron, SVM, etc.), is able to perfectly discriminate between a concept and one another in Figs. 6f and 7f. For instance, a line can separate the concept 1 (blue+magenta+cyan) from the non-concept 1 (red+green+yellow) in Fig. 7f. We test the SVM-classification capacity of the different set of vectors (Figs. 8 and 9). We can see on these figures that the error of classification is highly reduced by the concept vector learning, especially for the labeled set.

In the previous experiments, we assumed that there is no errors in the labels of Y. However, in real application

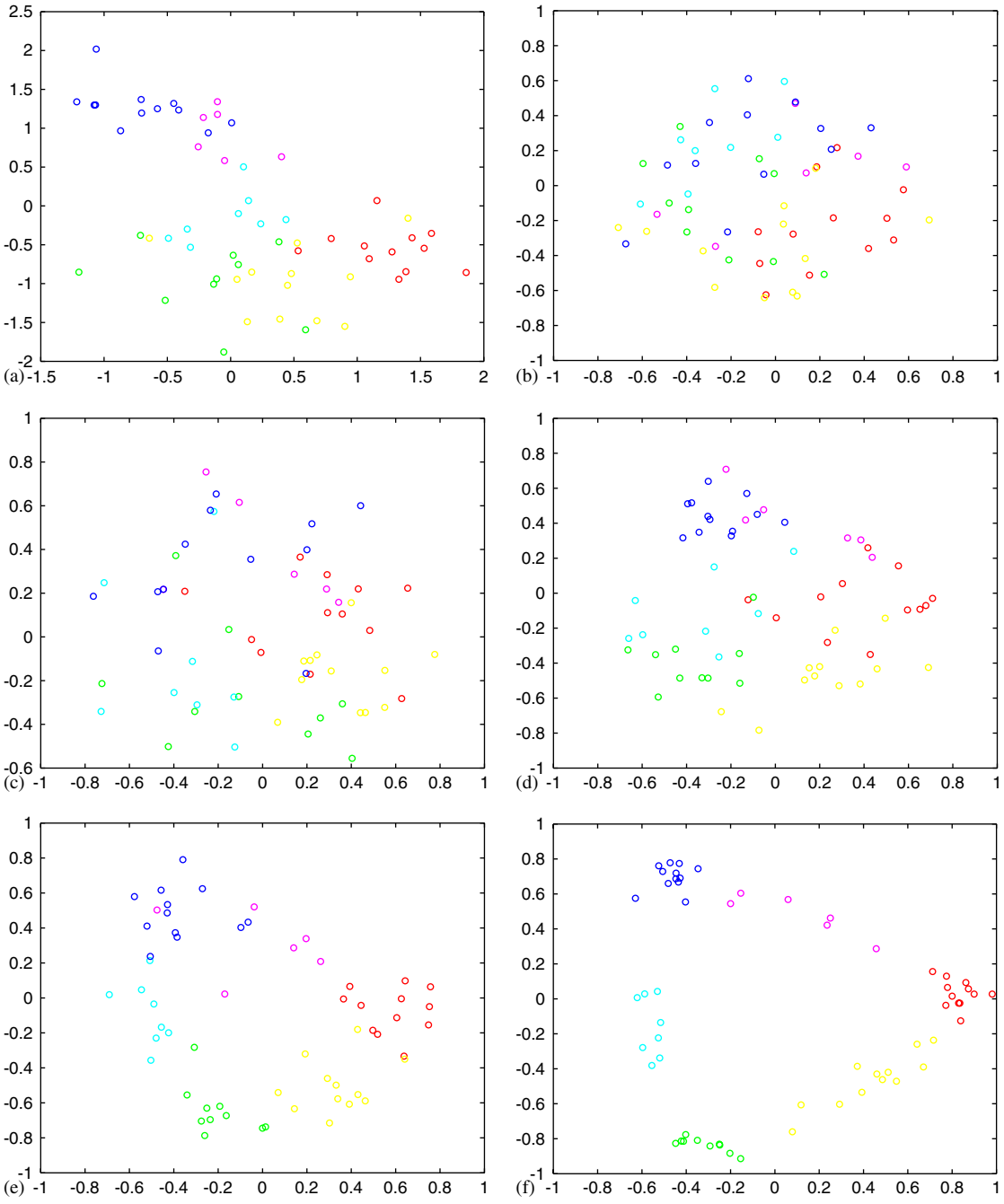

Fig. 10. Toy example with three mixed concepts. Legend: blue=concept 1, red=concept 2, green=concept 3, magenta=concepts 1 and 2, cyan=concepts 1 and 3, yellow = concepts 2 and 3. (a) Labeled set; (b) optimized labeled set, with 40% error; (c) optimized labeled set, with 30% error; (d) optimized labeled set, with 20% error; (e) optimized labeled set, with 10% error; (f) optimized labeled set, with no error.

erroneous labels can appear when data are collected from many user interaction.

In order to simulate these errors, we build new matrices of labels $Y^{10\%}$, $Y^{20\%}$, $Y^{30\%}$ and $Y^{40\%}$ based on the previous matrix Y, but including false labels. We initialize each new matrix with the matrix Y, and switch one non-zero labels in each $\mathbf{y}_p$ of $Y^{10\%}$, two non-zero labels in each $\mathbf{y}_p$ of $Y^{20\%}$, etc. As there is $N_l = 10$ non-zero values in each $\mathbf{y}_p$, switching 1 label represents an error rate of 10%, switching 2 labels represents an error rate of 20%, etc.

We present the labeled set and the various optimized labeled sets in Fig. 10. As the error rate increases, vectors are less gathered into clusters, and tend to be scattered in the space. This behavior is consistent with the idea of

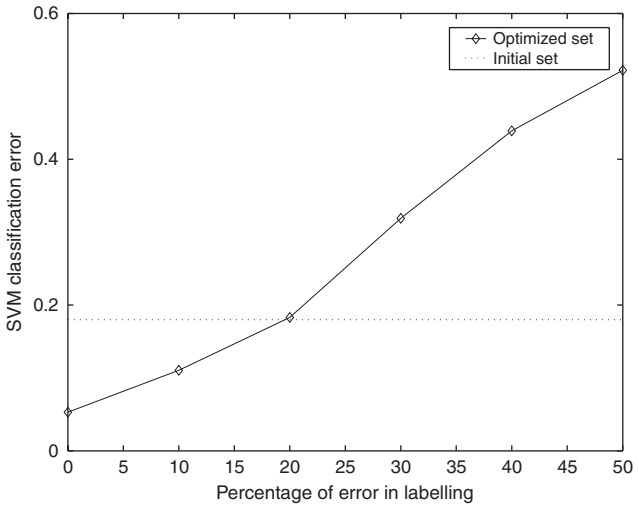

Fig. 11. Toy example with three mixed concepts. SVM-classification errors on the labeled set according to the percentage of error in labeling. The SVM-training set has 10 examples.

the method. The more there are erroneous labels, the more knowledge in Y is poor, but the method is still able to gather the vectors.

We also performed SVM-classification tests for each training sets with error. The error of classification for a SVM-training set of 10 examples are presented in Fig. 11. We included to this figure a dotted line which shows the error with the initial set. We can see that the more there are errors in a matrix Y, the more there are classification errors. On these experiments, our method is still efficient for an error rate lesser than 20%.

5.2. Toy example with $N_d > N_c - 1$

We build a toy example with 120 vectors of dimension $N_d = 3$ with $N_c = 3$ concepts (Fig. 12a). In this example, the dimension of vectors is not optimal according to Theorem 1. We also build set of labels Y compound of $N_y = 100$ labels $\mathbf{y}_p$ with $N_l = 10$ non-zero values. The labels are not necessarily balanced, a vector $\mathbf{y}_p$ can have more positive than negative values, and vice versa. The labeled set is compound of 60 vectors (Fig. 12b).

We can see in Fig. 12c and d, that after a concept vector learning with the proposed method, the vectors are distributed in a plane, and are as if they were in an optimal dimension. This result is due to the fact that equidistant centers in a higher space are also equidistant in a lower space, and the uniqueness of the distance remains true.

5.3. Real data

Tests are carried out on the generalist COREL photo database, which contains more than 50,000 pictures. To get tractable computation for the statistical evaluation, we randomly selected 77 of the COREL folders, to obtain a database of 6000 images. To perform interesting evaluation, we built from this database 50 concepts.[1] Each concept is built from 2 or 3 of the COREL folders. The concept sizes are from 50 to 300. The set of all the concepts covers the whole database, and many of them have common images with others.

We randomly build a set of labels Y simulating the use of the system. For instance, this set could be made from the labels given by users during the use of an image retrieval system. This set could also be made from text associated with each image. In all cases, we assume that the labels are incomplete, and have few non-zero values. In this context, we build labels with 25 positive values and 25 negative values, which is small (0.83%) against the size of the database (6000). Note that the method does not need balanced labels. The number of negative labels is usually large in comparison to the number of positive ones, and these last one must be large enough, for instance on this database the method need at least 10 positive labels per $\mathbf{y}_p$ to be efficient. We next train the concept vector learning algorithm with 250, 500 and 1000 partial labeling in Y. We obtain three new sets of vectors $X^{\star}_{250}$, $X^{\star}_{500}$ and $X^{\star}_{1000}$.

In order to evaluate the improvement, we experimented the SVM-classification capacity with a linear kernel of the new sets. Results are in Fig. 13. We drastically limited the size of the SVM-training set (up to 100 labels which represent 1.67% of the database size), to stay in a realistic CBIR context, where very few labels are usually available. Fig. 13 shows that the SVM-classification error decreases with the number of updates. With a set of 1000 labeling, the SVM-classification error is 4.6% with SVM-training set of 60 labels (1% of the database size), with no kernel optimization nor advanced learning improvements such as active learning. The gain is about 25% of less error, that means that starting with 1800 misclassified images, we obtain only 276 misclassified images with the new feature vectors.

Remark: The method needs 1 s to converge on a training set of $N_x = 6000$ vectors and $N_y = 1000$ partial labels with $N_l = 50$ non-zero values. It is particularly fast in comparison to related methods, as for instance the method in Ref. [10], which requires several minutes for the toy examples (with the matlab code they have published), and is untractable on sets with thousands of vectors.

6. Conclusion

In this paper, we introduce a concept vector learning method which improves the representation of a document collection, with few constraints on training data. The method is mostly based on the equidistance of concept centers,

[1] A description of this database and the 50 concepts can be found at: http://perso-etis.ensea.fr/~cord/data/mcorel50.tar.gz. This archive contains lists of image file names for all the concepts.

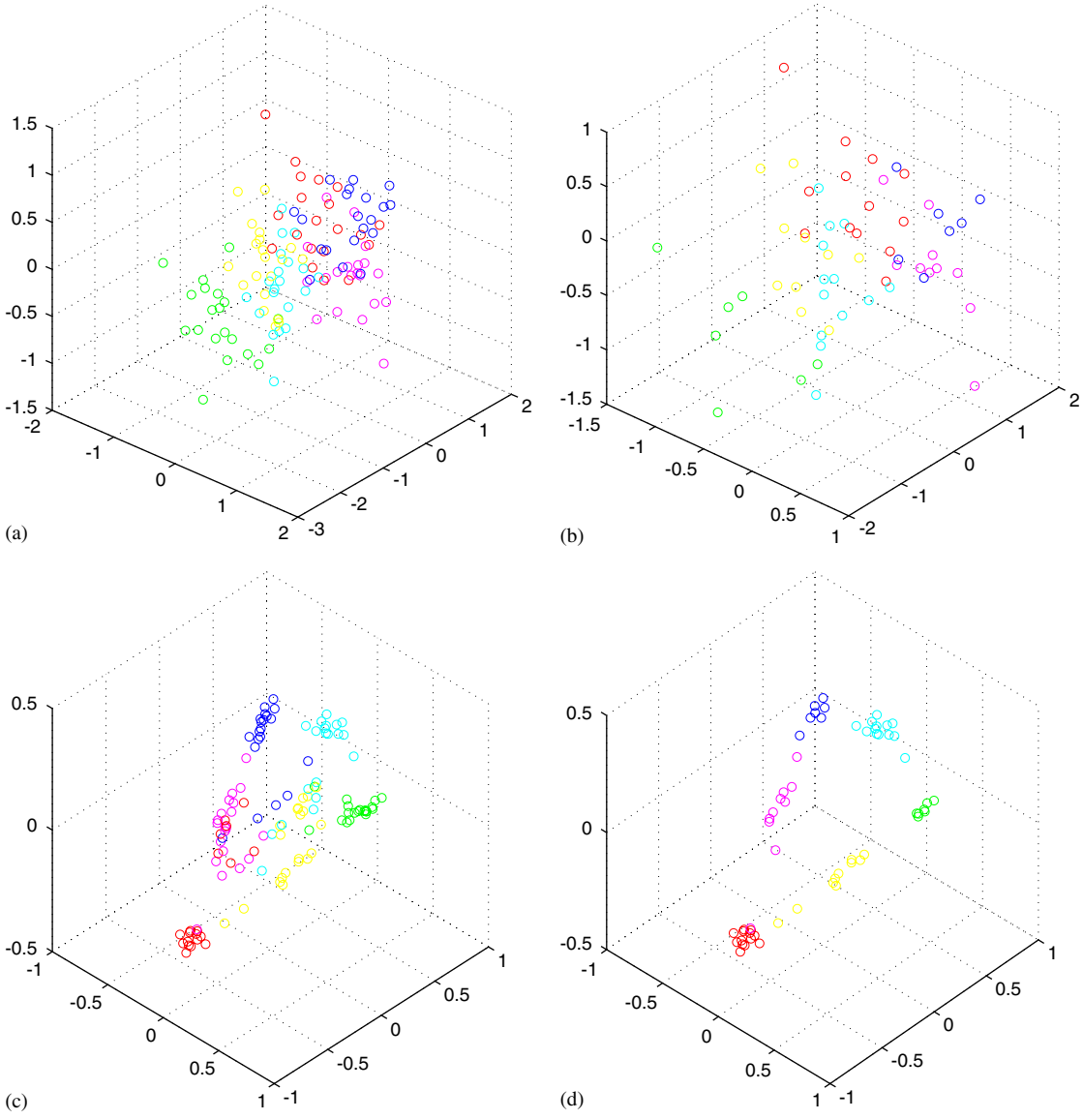

Fig. 12. Toy example with three mixed concepts in a 3D space. (a) Initial set; (b) labeled set; (c) optimized set with our method; (d) optimized labeled set with our method.

gathering the vectors of the same concept around each corresponding centers, and distributing the vectors in several concepts between those centers. Thus, the method is able to deal with mixed concepts, with the only constraint that the dimension of the vectors be superior to the number of concepts. We are actually working on an extension of this method to overcome this constraint, by applying the optimization in an infinite space, with a framework similar to the quasiconformal kernels approach [17,18]. If we assume that documents need to be gathered into concepts, then the method deals with a context where the size, the number and the structure of the concepts are unknown. Experiments carried out on synthetic and real data demonstrate the efficiency of the method. The representation of documents and the SVM classification are enhanced.

Note that the approach does not deal with a hierarchical organization of the data, when several concepts can form a "big" concept. However, the data distribution around equidistant centers is very interesting for post-processing. The simple fact that the data become equidistributed in an hypersphere allows new possibilities, for instance, using a K-means algorithm, one can compute an estimation of the concepts—their size, the membership of each vectors. Once the concepts are detected, powerful learning techniques (like in Ref. [1]) can be used in order to further improve the performances. As the data distribution allows concept detection, some adapted classification techniques could give good estimations of the membership of vectors, and then accurate representation of labeled and unlabeled data. Actually, application are many; to mention one, the management of a

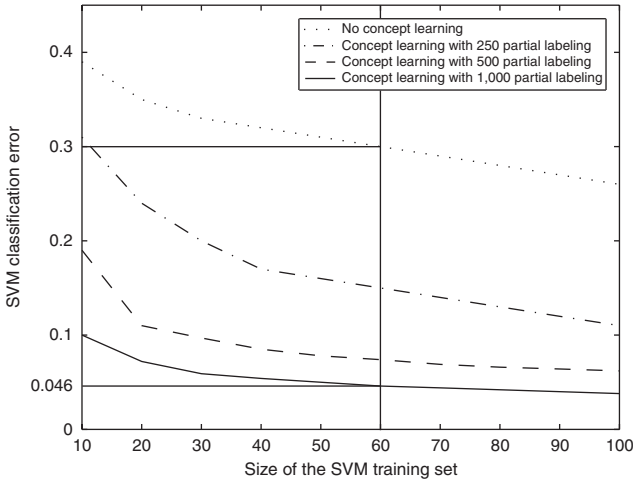

Fig. 13. Mean SVM-classification error for 50 concepts, according to the size of the SVM-training set. Each partial labeling has 25 positive values and 25 negative values.

collection of digital documents on the Internet. If the system is able to get semantic labels through its use, for instance when a user chooses a link in a search engine, or labels a picture, then the method can improve the overall performance of the system, during background optimization.

Appendix A. Proof of Theorem 1

Let $K = C^t C$ be the $N_c \times N_c$ matrix of all dot products between each vector of C. As $\forall j,\ j' \in 1 \dots N_c,\ \|\mathbf{c}_j\| = 1$ and $\|\mathbf{c}_j - \mathbf{c}_{j'}\|^2 = d$, it follows that:

$$\langle \mathbf{c}_j, \mathbf{c}_{j'} \rangle = \frac{1}{2}(\|\mathbf{c}_j\|^2 + \|\mathbf{c}_{j'}\|^2 - \|\mathbf{c}_j - \mathbf{c}_{j'}\|^2) = 1 - \frac{d}{2}.$$

If we set $h = 1 - d/2$, then K can be written as

$$K = \begin{pmatrix} 1 & h & h & \dots & h \\ h & 1 & h & \dots & h \\ h & h & \ddots & & \vdots \\ \vdots & \vdots & & 1 & h \\ h & h & \dots & h & 1 \end{pmatrix}.$$

K is the matrix of dot products between N_c vectors of dimension N_d, then rank $K \leqslant N_d$.

Now let us look for a condition to minimize the rank of K. In order to compute this rank, we compute the characteristic polynomial of K:

$$\det(K - \lambda I) = \det \begin{pmatrix} 1-\lambda & h & h & \dots & h \\ h & 1-\lambda & h & \dots & h \\ h & h & \ddots & & \vdots \\ \vdots & \vdots & & 1-\lambda & h \\ h & h & \dots & h & 1-\lambda \end{pmatrix}.$$

Let $\lambda = \lambda' - h + 1$, then

$$\begin{aligned} &\det(K - \lambda I) \\ &\quad = \det \begin{pmatrix} h-\lambda' & h & h & \dots & h \\ h & h-\lambda' & h & \dots & h \\ h & h & \ddots & & \vdots \\ \vdots & \vdots & & h-\lambda' & h \\ h & h & \dots & h & h-\lambda' \end{pmatrix} \\ &\quad = \det(h\mathbf{e}\mathbf{e}^t - \lambda' I) \end{aligned}$$

with $\mathbf{e}^t = (1 \dots 1)$.

The characteristic polynomial of the rank one matrix $h\mathbf{e}\mathbf{e}^t$ is

$$\det(h\mathbf{e}\mathbf{e}^t - \lambda' I) = (-\lambda')^{N_c - 1}(hN_c - \lambda').$$

Then, using $\lambda' = h - 1 + \lambda$:

$$\det(K - \lambda I) = ((1-h) - \lambda)^{N_c - 1}((1 + (N_c - 1)h) - \lambda).$$

The roots of the polynomial are $1 - h$ and $1 + (N_c - 1)h$.

The rank of K is minimized if we have the maximum number of zero roots. As we suppose that $d > 0$, then $h \neq 1$, and the first root cannot be zero.

In the case where $d = 2(1 + 1/(N_c - 1))$, $h = -1/(N_c - 1)$, and rank $K = N_c - 1$. Thus, $N_d \geqslant N_c - 1$.

In the case where $d \neq 2(1 + 1/(N_c - 1))$, rank $K = N_c$ and $N_d \geqslant N_c$.

References

[1] A. Natsev, M. Naphade, C.Y. Lin, J.R. Smith, Overcomplete representation and fusion for semantic concept detection, in: IEEE International Conference on Image Processing, Singapore, 2004.

[2] J. Rocchio, Relevance feedback in information retrieval, in: G. Salton (Ed.), The SMART Retrieval System: Experiments in Automatic Document Processing, Prentice-Hall Inc., New York, 1971.

[3] A. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, IEEE Trans. Pattern Anal. Mach. Intell. 22 (12) (2000) 1349–1380.

[4] S. Santini, A. Gupta, R. Jain, Emergent semantics through interaction in image databases, IEEE Trans. Knowl. Data Eng. 13 (3) (2001) 337–351.

[5] Y. Rui, T.S. Huang, M. Ortega, S. Mehrotra, Relevance feedback: a power tool in interactive content-based image retrieval, in: IEEE Transactions on Circuits and Systems for Video Technology, 1998, pp. 644–655.

[6] S. Tong, E. Chang, Support vector machine active learning for image retrieval, in: ACM Multimedia, 2001.

[7] P.H. Gosselin, M. Cord, RETIN AL: an active learning strategy for image category retrieval, in: IEEE International Conference on Image Processing, Singapore, 2004.

[8] H. Müller, W. Müller, D.M. Squire, S. Marchand-Maillet, T. Pun, Long-term learning from user behavior in content-based image retrieval, Technical Report, Computer Vision Group, University of Geneva, Switzerland, 2000.

[9] M. Schultz, T. Joachims, Learning a distance metric from relative comparisons, in: Neural Information Processing Systems, 2003.

[10] E.P. Xing, A.Y. Ng, M.I. Jordan, S. Russell, Distance metric learning, with application to clustering with side-information, in: Neural Information Processing Systems, Vancouver, British Columbia, 2002.

[11] J. Fournier, M. Cord, Long-term similarity learning in content-based image retrieval, in: International Conference in Image Processing (ICIP), Rochester, New York, USA, 2002.

[12] J. Han, M. Li, H. Zhang, L. Guo, A memorization learning model for image retrieval, in: IEEE International Conference on Image Processing, Barcelona, Spain, 2003.

[13] N. Cristianini, J. Shawe-Taylor, A. Elisseff, J. Kandola, On kernel target alignment, in: Neural Information Processing Systems, Vancouver, Canada, 2001.

[14] G.R.G. Lanckriet, N. Cristianini, N. Bartlett, L. ElGhaoui, M.I. Jordan, Learning the kernel matrix with semi-definite programming, in: International Conference on Machine Learning, Sydney, Australia, 2002.

[15] D.R. Heisterkamp, Building a latent semantic index of an image database from patterns of relevance feedback, in: International Conference on Pattern Recognition, vol. 4, Quebec City, Canada, 2002, pp. 132–137.

[16] P.H. Gosselin, M. Cord, Semantic kernel learning for interactive image retrieval, in: IEEE International Conference on Image Processing, Genova, Italy, 2005.

[17] S. Amari, S. Wu, Improving support vector machine classifiers by modifying kernel functions, Neural Networks 12 (6) (1999) 783–789.

[18] D.R. Heisterkamp, J. Peng, H.K. Dai, Adaptive quasiconformal kernel metric for image retrieval, in: International Conference on Computer Vision and Pattern Recognition, 2001.