Pattern Anal Applic (2008) 11:227–246
DOI 10.1007/s10044-007-0082-x

THEORETICAL ADVANCES

# Distance-based discriminant analysis method and its applications

Serhiy Kosinov · Thierry Pun

Received: 29 November 2006 / Accepted: 10 July 2007 / Published online: 11 September 2007

**Abstract** This paper proposes a method of finding a discriminative linear transformation that enhances the data's degree of conformance to the compactness hypothesis and its inverse. The problem formulation relies on inter-observation distances only, which is shown to improve non-parametric and non-linear classifier performance on benchmark and real-world data sets. The proposed approach is suitable for both binary and multiple-category classification problems, and can be applied as a dimensionality reduction technique. In the latter case, the number of necessary discriminative dimensions can be determined exactly. Also considered is a kernel-based extension of the proposed discriminant analysis method which overcomes the linearity assumption of the sought discriminative transformation imposed by the initial formulation. This enhancement allows the proposed method to be applied to non-linear classification problems and has an additional benefit of being able to accommodate indefinite kernels.

**Keywords** Discriminant analysis · Feature extraction · Iterative majorization · Content-based image retrieval

## 1 Originality and contribution

In this paper we focus on finding a transformation of the data that forces it to conform to the compactness hypothesis and its inverse. Relying exclusively on distances among the observations, we set up the task of deriving a discriminative transformation as a problem of optimizing a criterion whose formulation is motivated by the ideas of version space center methods. The optimization problem, in turn, is solved via the technique of iterative majorization. Once the discriminative transformation has been found and applied to the data, we use nearest neighbor classification as well as other non-parametric approaches to distinguish among different classes of observations.

The main advantages of the proposed approach are its suitability for both binary and multiple-category discriminant analysis problems, the flexibility of the formulation that renders the method as a dimensionality reduction technique, the ability to determine the necessary dimensionality of the discriminative transformation, and its non-parametric nature that lets the technique work well for non-Gaussian data. These and other essential properties of the proposed method are discussed in comparison with relevant techniques, such as principal component analysis, linear discriminant analysis, biased discriminant analysis, discriminant adaptive nearest neighbor, non-parametric discriminant analysis, etc. In addition to that, we also consider a kernel-based extension of the proposed discriminant analysis method thereby overcoming the limiting linearity assumption of the sought discriminative transformation imposed by the initial formulation. Performance tests of the proposed method on a number of standard UCI benchmark data sets and in the application to image retrieval show a favorable improvement in classification accuracy.

## 2 Introduction

This article describes a method for finding a discriminative transformation based on the compactness hypothesis [1] and motivated by the ideas of the version space center

S. Kosinov (✉) · T. Pun
24, rue General-Dufour, 1211 Geneva, Switzerland
e-mail: kosinov@cui.unige.ch

methods. The proposed method specifically aims at improving the accuracy of the non-parametric type of classifiers, such as nearest neighbor (NN) [17], and is sought to have the following characteristics:

- ability to perform discriminative *feature extraction* and *dimensionality reduction*, while possessing the means to determine how many dimensions are sufficient to distinguish among a given set of classes,
- *assymetry of formulation* suitable for the most popular deployment scenarios in 1-against-all classification, as well as in the case of data set imbalance,
- *transformational* and *non-parametric* method specification that would allow for extensions, use as a discriminative data pre-processing technique, minimal assumptions on data distribution, and maximum utilization of the capabilities of the prospective classifier to be used with the transformed data,
- ease of extension to a *non-linear* problem setting via kernels, as well as *multiple-category* case.

The following sections provide a detailed account of the proposed method and the various aspects relating to its formulation, algorithmic specification, numerical implementaion, extensions, and experimental evaluation. We deliberately defer the comparison of the proposed method with other relevant techniques until Sect. 5 in an express effort to provide a more thorough and deatiled discussion later on.

## 3 Problem formulation

Suppose that we seek to distinguish between two classes represented by data sets $X$ and $Y$ having $N_X$ and $N_Y$ $m$-dimensional observations, respectively. For this purpose, we are looking for such transformation matrix $T \in \mathbb{R}^{m \times k}, k \ll m$, such that $\{X \mapsto X', Y \mapsto Y'\}$, that places instances of a given class near each other while relocating the instances of the other class sufficiently far away. In other words, we want to ensure that the compactness hypothesis [1] holds for either of the two classes in question, while its opposite is true for both.

While the above preamble may fit just about any class-separating discriminant analysis method profile (e.g., [7, 14, 21, 27, 38, 59]), we must emphasize several important assertions that distinguish the presented method and naturally lead to the problem formulation that follows. First of all, we must reiterate that one of our primary goals is to improve the performance of a non-parametric classifier, such as NN. Therefore, the sought problem formulation must relate only to the factors that directly influence the decisions made by the classifier, such as the distances among observations. Secondly, in order to benefit as much as possible from the non-parametric nature of the NN, the sought formulation must not rely on the traditional class separability and scatter measures that use class means, weighted centroids or their variants [20] which, in general, connote quite strong distributional assumptions. Finally, an asymmetric product form should be more preferable, justified as consistent with the properties of the data encountered in many target application areas, such as content-based image retrieval and categorization [63], as well as beneficial from the viewpoint of insightful parallels to some version space center methods discussed later in this section.

Let $d_{ij}^W(T)$ denote a Euclidean distance between observations $i$ and $j$ from transformed data set $X'$ given a transformation matrix $T$, and, analogously, $d_{ij}^B(T)$ specify a distance between the $i$th observation from data set $X'$ and the $j$th observation from data set $Y'$, where superscripts "$W$" and "$B$" stand for within-class and between-class type of distance, respectively:

$$d_{ij}^W = \sqrt{(x_i - x_j)^{\mathrm{T}} T T^{\mathrm{T}} (x_i - x_j)}, \tag{1}$$

$$d_{ij}^B = \sqrt{(x_i - y_j)^{\mathrm{T}} T T^{\mathrm{T}} (x_i - y_j)}, \tag{2}$$

for $\{x_i\}_{i=1}^{N_X} \in \mathbb{R}^m, \{y_j\}_{j=1}^{N_Y} \in \mathbb{R}^m$. Using this notation, the sought discriminative data transformation can be obtained by minimizing the following criterion:[1]

$$J(T) = \frac{\left(\prod_{i<j}^{N_X} \Psi\left(d_{ij}^W(T)\right)\right)^{\frac{2}{N_X(N_X-1)}}}{\left(\prod_{i=1}^{N_X}\prod_{j=1}^{N_Y} d_{ij}^B(T)\right)^{\frac{1}{N_X N_Y}}}, \tag{3}$$

where the numerator and denominator of (3) represent the geometric means of corresponding distances, and $\Psi(d_{ij}^W(T))$ denotes a Huber robust estimation function [29] parametrized by a positive constant $c$ and defined as:

$$\Psi(d_{ij}^W) = \begin{cases} \frac{1}{2}\left(d_{ij}^W\right)^2 & \text{if } d_{ij}^W \le c; \\ c d_{ij}^W - \frac{1}{2}c^2 & \text{if } d_{ij}^W > c. \end{cases} \tag{4}$$

The choice of Huber function in (3) is motivated by the fact that at $c$ the function switches from quadratic to linear penalty allowing to mitigate the consequences of an implicit unimodality assumption that the formulation of the numerator of (3) leads to. Additionally, Huber function has several attractive properties, such as strong convexity and bounded second derivative, that greatly facilitate the derivation of the majorizing inequalities, as will be shown in Sect. 3.2.

[1] Here and in several other places we will use shorthand $\prod_{i<j}^{N_X}$ to designate double product $\prod_{i=1}^{N_X}\prod_{j=i+1}^{N_X}$.

In the logarithmic form, criterion (3) is written as:

$$\begin{aligned}\log J(T) &= \frac{2}{N_X(N_X-1)}\sum_{i<j}^{N_X}\log \Psi\left(d_{ij}^W(T)\right)\\ &\quad - \frac{1}{N_X N_Y}\sum_{i=1}^{N_X}\sum_{j=1}^{N_Y}\log d_{ij}^B(T)\\ &= \alpha S_W(T) - \beta S_B(T),\end{aligned} \tag{5}$$

which highlights the theoretical underpinnings motivating the above formulation. Indeed, the between-class part of log $J(T)$, being a weighted sum of log-barrier functions [42], may be viewed as an extended formulation of analytic center machine (ACM) method that finds a separating hyperplane as an analytic center of the classifier version space [55]. Namely, by setting the transformation matrix $T$ to be a column vector defining some separating hyperplane, (5) may be shown to include an approximation of the objective of the ACM method. Thus the above formulation has a valuable connection to the version space center methods, yet manages to avoid some of their disadvantages.[2]

For notational convenience, the first and the second summation terms of (5) are going to be referred to as $S_W(T)$ (“within” distances) and $S_B(T)$ (“between” distances) in the following discussion to allow for a more convenient notation and due to their apparent functional similarity with the notions of within- and between-class scatter measures used in a number of well-known discriminant analysis techniques [14, 16, 21, 27]. We will also shorten the notation by reassigning the normalizing quantities $\frac{2}{N_X(N_X-1)}$ and $\frac{1}{N_X N_Y}$ to $\alpha$ and $\beta$, respectively.

Although a straightforward differentiation of (5) might appear sufficient in order to proceed with a generic optimization search technique such as gradient descent, our preliminary experiments showed that the quality of the found solutions is severely impaired by the problems due to local minima and considerable degree of dependence on the initial starting value, as detailed in Sect. 7. Moreover, the computational costs of such an endeavor very quickly become prohibitive and are further exacerbated if, in addition to the descent direction, a proper step length must be calculated, so that gradient descent does not overshoot and actually manages to improve the optimization criterion, while the latter outcome is guaranteed by the introduced below iterative majorization technique (and, hence its alternative name: “guaranteed descent”). Furthermore, some of the tested state-of-the-art optimization routines, such as SQP and Quasi-Newton with line search, did not scale well either and happened not to be able to converge, even on fairly simple data sets.

In order to avoid the above pitfalls, it was decided to derive some useful approximations of criterion (5) that would make the task of its optimization amenable to a straightforward procedure based on the iterative majorization method, which we discuss in the following section.

## 4 Iterative majorization

### 4.1 General overview of the method

As stated in [6, 28, 56], the central idea of the majorization method is to replace the task of optimizing a complicated objective function $f(x)$ by an iterative sequence of simpler minimization problems in terms of the members of the family of auxiliary functions $\mu(x,\bar{x})$, where $x$ and $\bar{x}$ vary in the same domain $\Omega$. In order for $\mu(x,\bar{x})$ to qualify as a *majorizing function* of $f(x)$, the auxiliary function $\mu(x,\bar{x})$ is required to fulfill the following conditions, for $x,\bar{x}\in\Omega$ :

- the auxiliary function $\mu(x,\bar{x})$ should be simpler to minimize than $f(x)$,
- the original function must always be less or equal to the auxiliary function:

$$f(x)\le\mu(x,\bar{x}), \tag{6}$$

- the auxiliary function should touch the surface of the original function at the *supporting point*[3] $\bar{x}$ :

$$f(\bar{x})=\mu(\bar{x},\bar{x}). \tag{7}$$

To understand the principle of minimizing a function by majorization, consider the following observation [6]. Let the minimum of $\mu(x,\bar{x})$ over $x$ be attained at $x^*$. Then, (6) and (7) imply the chain of inequalities

$$f(x^*)\le\mu(x^*,\bar{x})\le\mu(\bar{x},\bar{x})=f(\bar{x}). \tag{8}$$

This chain of inequalities is named the *sandwich inequality* by De Leeuw [33], because the minimum of the majorizing function $\mu(x^*,\bar{x})$ is squeezed between $f(x^*)$ and $f(\bar{x})$. A graphic illustration of these inequalities is shown in Fig. 1 for two subsequent iterations of iterative majorization of function $f(x)$. Thus, given an appropriate function $\mu(x,\bar{x})$, the iterative majorization (IM) algorithm proceeds as follows:

[2] For instance, in contrast to the ACM technique, the DDA formulation applies naturally to the cases where there is no strict class separability, whereas the ACM method fails because the version space becomes an empty set.

[3] The similar notation will be used further on, where a dash over a variable name will signify that the variable either depends on or is itself a supporting point.

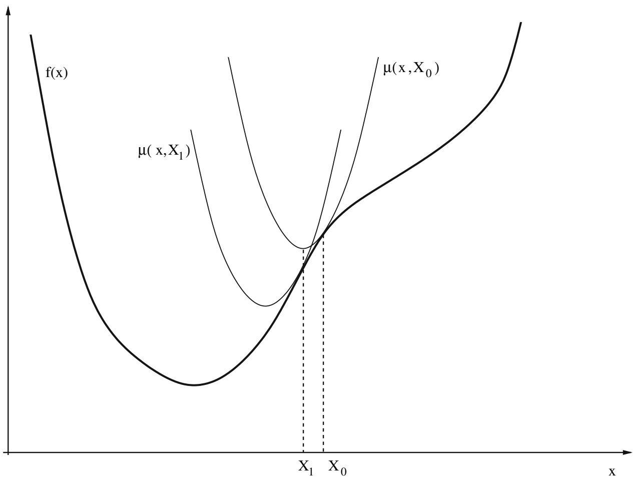


**Fig. 1** Illustration of two subsequent iterations of the iterative majorization method. The first iteration starts by finding the auxiliary function $\mu(x, X_0)$, which is located above the original function $f(x)$ and touches at the supporting point $X_0$. The minimum of the auxiliary function $\mu(x, X_0)$ is attained at $X_1$, where $f(X_1)$ can never be larger than $\mu(X_1, X_0)$. This completes one iteration. The second iteration proceeds analogously from supporting point $X_1$, and so on, until convergence

1. Assign an initial supporting point $\bar{x} = \bar{x}_0 \in \Omega$, choose tolerance $\varepsilon$;
2. Find a successor point $x_s : x_s = \arg\min_{x \in \Omega} \mu(x, \bar{x})$;
3. If $f(\bar{x}) - f(x_s) < \epsilon$, then stop;
4. Set $\bar{x} = x_s$, go to 2.

The essential property of the above procedure is that it generates a non-increasing sequence of function values, which converges to a stationary point whenever $f(x)$ is bounded from below and $x$ is sufficiently restricted. As noted by Fletcher [18], the found point is in most cases a local minimizer. Furthermore, according to the results reported by Van Deun et al. [56], the majorization method has a valuable property of a low to negligible dependence on the initial value, compared to other applicable techniques. Another advantage of the majorization approach is due to the fact that there exist a number of specifically tailored global optimization techniques, such as objective function tunneling [6], that can be applied if the problem domain is abundant with low quality local minima. In the next section we will derive the majorizing expressions of (5) and show how they are used for optimizing the chosen criterion.

### 4.2 Majorizing the optimization criterion

It can be verified that majorization remains valid under additive decomposition. Therefore, a possible strategy for majorizing (5) is to deal with $S_W(T)$ and $-S_B(T)$ separately and subsequently recombine their respective majorizing expressions.

We begin by noting that the logarithm, as much as any other concave function, can always be majorized by a straight line $y = ax + b$ whose coefficients $a = 1/\bar{x}$ and $b = \log(\bar{x}) - 1$ are determined from the majorization requirements (6) and (7) rendering

$$\log(x) \le \bar{x}^{-1}x + \log(\bar{x}) - 1. \tag{9}$$

Also, as previously reported in [9, 28], Huber distance (4) is convex and has a bounded second derivative, and hence can be majorized by a convex quadratic function:

$$\Psi(x) \le \frac{1}{2}\bar{w}x^2 + \frac{1}{2}(\bar{v} + sign(\bar{x} - c)\bar{v}), \tag{10}$$

where $x > 0$, and coefficients $\bar{v}$ and $\bar{w}$ are defined as:

$$\bar{v} = \frac{1}{2}c\bar{x} - \frac{1}{2}c^2, \tag{11}$$

$$\bar{w} = \begin{cases} 1 & \text{if } \bar{x} \le c; \\ \frac{c}{\bar{x}} & \text{if } \bar{x} > c. \end{cases} \tag{12}$$

Combining (9) and (10) together while substituting the result into the formulation of $S_W(T)$, we can obtain its majorizing expression $\mu_{S_W}(T, \bar{T})$ :

$$\begin{aligned} S_W(T) &= \sum_{i<j}^{N_X} \log \Psi\left(d_{ij}^W(T)\right) \\ &\le \sum_{i<j}^{N_X} \frac{\bar{w}_{ij} \cdot \left(d_{ij}^W(T)\right)^2}{2\Psi\left(d_{ij}^W(\bar{T})\right)} + K_1 \\ &= \mu_{S_W}(T, \bar{T}), \end{aligned} \tag{13}$$

where $T, \bar{T} \in \mathbb{R}^{m \times m}, \bar{T}$ is a supporting point for $T, \bar{w}_{ij}$ is a weight of the Huber function majorizer, that in this case is equal to 1 if $\Psi(d_{ij}^W(\bar{T})) < c$ or $c/\Psi(d_{ij}^W(\bar{T}))$ otherwise, and $K_1$ is a constant term that collects all of the other terms that are irrelevant from the point of view of minimization with respect to $T$. Switching to matrix notation (see "Appendix" for derivation details), we define a square symmetric matrix $R$:

$$r_{ij} = \begin{cases} -\frac{\bar{w}_{ij}}{\Psi\left(d_{ij}^W(\bar{T})\right)} & \text{if } i \ne j; \\ -\sum\limits_{k=1, k \ne i}^{N_X} r_{ik} & \text{if } i = j; \end{cases} \tag{14}$$

which lets us rewrite the majorizing expression of $S_W(T)$ in its final form, as follows:

$$\mu_{S_W}(T, \bar{T}) = \frac{1}{2}\mathbf{tr}\left(T^{\mathrm{T}}X^{\mathrm{T}}RXT\right) + K_1. \tag{15}$$

An attempt to majorize $-S_B(T)$ directly runs into problems due to the difficulties of finding a proper quadratic majorizing function of the negative logarithm. As a practical solution we consider two alternative replacements of –log($x$) in $-S_B(T)$:

- a piece-wise linear approximation,
- a second order Taylor expansion.

According to the first alternative, we replace the neg-logarithm with its piece-wise linear approximation (see an illustration in Fig. 2), which, in turn, can be represented as a sum of the functions defined as:

$$g(x;x_0,l,r)=\begin{cases} r(x-x_0) & \text{if } x \geq x_0,\\ -l(x-x_0) & \text{if } x < x_0; \end{cases} \tag{16}$$

where $l + r > 0$, to ensure convexity. It is easy to see that the family of functions defined in (16) is one of the many possible generalizations of the absolute value function |$x$|, the former being equivalent to the latter whenever $x_0 = 0$ and $l = r = 1$. Similarly to |$x$|, $g(x;x_0,l,r)$ can be majorized by a quadratic $ax^2 + bx + c$ with coefficients

$$a=\frac{r+l}{4|\bar{x}-x_0|}, \tag{17}$$

$$b=\frac{r-l}{2}-\frac{(r+l)x_0}{2|\bar{x}-x_0|}, \tag{18}$$

$$c=\frac{(r+l)x_0^2}{4|\bar{x}-x_0|}+\frac{(l-r)x_0}{2}+\frac{(r+l)|\bar{x}-x_0|}{4}, \tag{19}$$

for a supporting point $\bar{x}$ and $a > 0$, $b$ and $c$ determined directly from the majorization requirements (6) and (7). Figure 3 depicts an example of a function from $g(x;x_0,l,r)$ family alongside its majorizer. The final expression of the majorizer based on the piece-wise linear approximation, as derived by carrying out calculations similar to those given in "Appendix", is quite unwieldy and computationally costly even for a moderate number of $g$-family functions comprising the approximation. For this reason, we chose the other solution provided by a Taylor series expansion, as a faster and more stable alternative.[4]

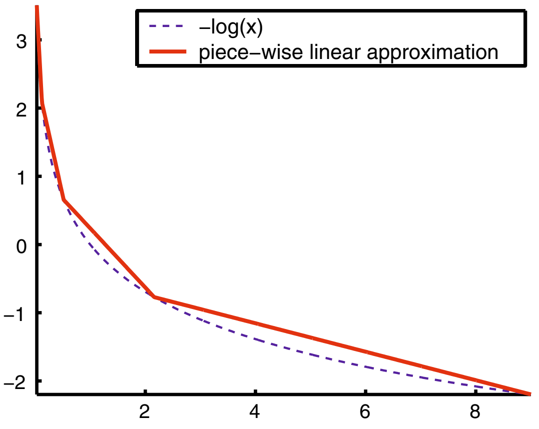


**Fig. 2** Piece-wise linear approximation of –log($x$)

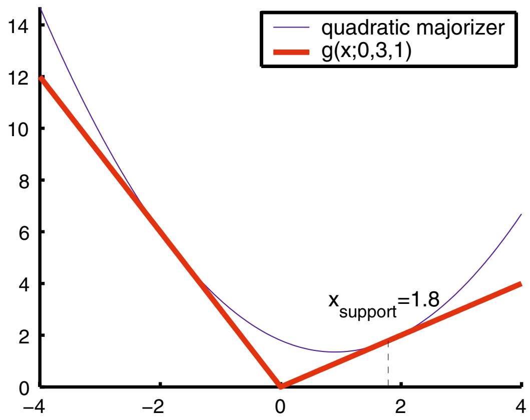


**Fig. 3** Example of a quadratic majorizer of $g(x;0,\ 3,\ 1)$ around supporting point $\bar{x} = 1.8$

Following the second approach, we express every term of $S_B(T)$ using a second order Taylor series expansion of the logarithm function around a supporting point $\bar{T}$ :

$$\log\left(d_{ij}^B(T)\right)\approx-\frac{1}{2}\left(\frac{d_{ij}^B(T)}{d_{ij}^B(\bar{T})}\right)^2+2\frac{d_{ij}^B(T)}{d_{ij}^B(\bar{T})}+\log\left(d_{ij}^B(\bar{T})\right)-\frac{3}{2}. \tag{20}$$

Substituting (20) into the expression of $-S_B(T)$ leads to:

$$\begin{aligned}-S_B(T)&=-\sum_{i=1}^{N_X}\sum_{j=1}^{N_Y}\log d_{ij}^B(T)\\ &\approx\frac{1}{2}\sum_{i=1}^{N_X}\sum_{j=1}^{N_Y}\left(\frac{d_{ij}^B(T)}{d_{ij}^B(\bar{T})}\right)^2\\ &\quad-2\sum_{i=1}^{N_X}\sum_{j=1}^{N_Y}\frac{d_{ij}^B(T)}{d_{ij}^B(\bar{T})}+K_2,\end{aligned} \tag{21}$$

where $K_2$ is a constant term that collects all of the other terms that are irrelevant from the point of view of minimization with respect to $T$. One may notice that in (21) only the second term, the sum of appropriately scaled negative Euclidean distances, requires majorization since the other two are either constant with respect to $T$ or given as a quadratic which is simple enough to handle as is.

In order to find a majorizing expression of (21) we will make use of a well-known fact frequently mentioned in

[4] A more detailed analysis may demonstrate that resorting to the Taylor series approximation might break conformance to the majorization requirements in the strict sense. However, the empirical evidence proved otherwise (see section 7), confirming the technique as an alternative of preference.

literature [6, 9, 28, 56], stating that the negative of a Euclidean distance is linearly majorizable:

$$-||x|| \leq -\frac{\bar{x}^{\mathrm{T}}x}{||\bar{x}||} \tag{22}$$

which is a direct consequence of the Cauchy–Schwarz inequality $||x||||\bar{x}|| \geq \bar{x}^{\mathrm{T}}x$. Switching to matrix notation (see "Appendix" for derivation details), we define a square symmetric matrix $G$ of size $N = N_X + N_Y$, such that:[5]

$$g_{ij} = \begin{cases} -\frac{1}{\left(d_{ij}^{B}(\bar{T})\right)^2} & \text{for } i \in [1; N_X] \\ & \text{and } j \in [N_X + 1; N], \\ -\frac{1}{\left(d_{ij}^{B}(\bar{T})\right)^2} & \text{if } i \in [N_X + 1; N] \\ & \text{and } j \in [1; N_X], \\ -\sum\limits_{k=1, k \neq i}^{N_X+N_Y} g_{ik} & \text{if } i = j, \end{cases} \tag{23}$$

which, combined with the result of (22) substituted into (21), lets us derive the majorizing expression for $-S_B(T)$ in its final form, as follows:

$$\begin{aligned} \mu_{-S_B}(T, \bar{T}) = {} & \frac{1}{2}\mathbf{tr}(T^{\mathrm{T}}Z^{\mathrm{T}}GZT) \\ & - 2\mathbf{tr}(T^{\mathrm{T}}Z^{\mathrm{T}}GZ\bar{T}) + K_2, \end{aligned} \tag{24}$$

where $Z$ is the matrix obtained by joining $X$ and $Y$ together, row-wise:

$$Z = \begin{bmatrix} X \\ Y \end{bmatrix}. \tag{25}$$

Finally, combining results (15) and (24), we obtain a majorizing function of the log $J(T)$ optimization criterion:

$$\begin{aligned} \mu_{\log J}(T, \bar{T}) & = \alpha\mu_{S_W} + \beta\mu_{-S_B} \\ & = \frac{\alpha}{2}\mathbf{tr}\left(T^{\mathrm{T}}X^{\mathrm{T}}RXT\right) \\ & \quad + \frac{\beta}{2}\mathbf{tr}(T^{\mathrm{T}}Z^{\mathrm{T}}GZT) \\ & \quad - 2\beta\mathbf{tr}(T^{\mathrm{T}}Z^{\mathrm{T}}GZ\bar{T}) + K_3, \end{aligned} \tag{26}$$

that can be used to find an optimal transformation $T$ minimizing log $J(T)$ criterion via the iterative procedure outlined in Sect. 3.1. Similarly to the expressions shown in (13) and (21), $K_3$ is a constant term that collects all of the other terms that are irrelevant from the point of view of minimization with respect to $T$.

### 4.3 Minimization of the majorizer of log $J(T)$

It is possible to minimize (26) with respect to $T$ in a straightforward fashion by setting its derivative to zero and solving the resulting system of linear equations with any of the computationally efficient methods, such as QR decomposition [22]. However, it is often recommended [3, 31, 32] that a length-constrained (or, regularized, as usually referred to in the domains of signal processing, inverse problems [4] and regularized risk minimization [57]) solution be found by deploying such techniques as weight-limiting, weight decay, etc., especially in the case of classifiers capable of achieving zero training error, to prevent overfitting and thus improve generalization performance of the classifier. In order to find an optimal transformation $T$ that satisfies the length constraint, we first form the Lagrangian function

$$\mathcal{L} = \mu_{\log J}(T, \bar{T}) + \lambda(\mathbf{tr}(T^{\mathrm{T}}T) - \Delta), \tag{27}$$

where $\lambda$ is a Langrangian multiplier and $\Delta$ is the value of the length constraint that is estimated from the classification performance on a validation data set [39]. It follows from (27) that an optimal solution $T$ is:

$$T = (M + 2\lambda I)^{-1}L \tag{28}$$

where $M$ is defined as $\frac{\alpha}{\beta}X^{\mathrm{T}}RX + Z^{\mathrm{T}}GZ, L$ is equal to $2Z^{\mathrm{T}}GZ\bar{T}$, and $I$ is an identity matrix. Plugging (28) back into the expression of the length constraint, we obtain the following:

$$\begin{aligned} \Delta & = \mathbf{tr}\left(L^{\mathrm{T}}(M + 2\lambda I)^{-1}(M + 2\lambda I)^{-1}L\right) \\ & = \mathbf{tr}\left(L^{\mathrm{T}}U\frac{1}{(2\lambda I + D)^2}U^{\mathrm{T}}L\right). \end{aligned} \tag{29}$$

where $U$ and $D$ are the respective matrices of eigenvectors and eigenvalues of $M$. Here, we have used the fact that symmetric matrices $M$ and $M + 2\lambda\ I$ have the same eigenvectors, while the eigenvalues of $M + 2\lambda\ I$ are equal to those of $M$ increased by $2\lambda$. Also, to simplify the notation of (29), the reciprocal and squaring operations should be understood as applied to the diagonal matrix $D$ on the element by element basis taking into account the magnitudes of each eigenvalue so as to avoid division by zero problems. Clearly (29), is an equation of one variable $\lambda$ with a computable derivative, that is easily solved by any suitable root-finding technique, such as Newton-Raphson method, or with a method specifically tailored to solving this type of problems, commonly referred to as a TRS, i.e. *trust region problem* [40, 46]. Once the constraint-satisfying value $\lambda$ has been found, the optimal transformation $T$, i.e. the successor point in the iterative majorization algorithm is recovered as:

[5] The elements $g_{ij}$ of matrix $G$ not affected by the first two rules of (23) are assumed to have been initially set to zero.

$$T_s = U(2\lambda I + D)^{-1}U^{\mathrm{T}}L, \tag{30}$$

where the bracketed expression is a diagonal matrix whose inverse is easily computed through the reciprocal of the diagonal elements.

It should be mentioned that for the problems such as minimization of (26) the universally suggested approach [28, 30] is to decompose the design matrices of each quadratic component of the function being optimized into a sum of a diagonal positive definite and a negative definite matrices, and use the definiteness property to derive another majorizing inequality. This method, although theoretically sound and well-justified, in our experiments demonstrated a significantly slower rate of convergence induced by larger condition number of the matrices involved, and thus was subsequently replaced by the solution defined in (30), even though the latter method involves a costly eigendecomposition operation.

## 5 Putting it all together

### 5.1 Complete algorithm

Considering all of the derivations we have desribed so far, the complete distance-based discriminant analysis (DDA) algorithm for iterative majorization of log $J(T)$ criterion (5) can be specified as follows:

**Algorithm DDA**.

```
1. Assign an initial supporting point $\bar{T} = \bar{T}_0 \in \mathbb{R}^{m\times m}$;
2. Find a successor point $T_s$ using (30);
3. If $\log J(\bar{T}) - \log J(T_s) < \epsilon$, then stop;
4. Set $\bar{T} = T_s$, go to 2.
```

### 5.2 Dimensionality reduction

Observe that setting the column size of $T$ to an arbitrary value $k \ll m$ renders the presented method of DDA a dimensionality reduction technique[6] that may be used in a variety of applications such as feature selection, low-dimensional data visualization, etc. Moreover, the value of $k$, i.e., the exact number of dimensions the data can be reduced to without loss of discriminatory power with respect to (5), is precisely determined by the number of non-zero singular values of $T$. Indeed, the distances between the transformed observations may be viewed as distances between the original observations in a different metric $TT^{\mathrm{T}}$, that can be expressed as $TT^{\mathrm{T}} = USV^{\mathrm{T}}VSU^{\mathrm{T}} = U_kS_k^2U_k^{\mathrm{T}}$ using the singular value decomposition of $T$. The obtained expression reveals that the effect of the full-dimensional transformation $T$ is captured by the first $k$ left-singular vectors of $T$ scaled by the corresponding non-zero singular values, whose number gives an answer to the question of how many dimensions are needed in the transformed space.

A summary of various other properties that distinguish DDA from existing dimensionality reduction methods is provided in Sect. 5.

### 5.3 Multiple class discriminant analysis

While the above discussion is concentrated mostly on the two-class configuration, it is straightforward to generalize the presented formulation to a multiple-class discriminant analysis setting, for the number of classes $K \geq 2$:

$$\log J_K(T) = \sum_{i=1}^{K-1}\left(\alpha^{(i)}S_W(T)^{(i)} - \beta^{(i)}S_B(T)^{(i)}\right), \tag{31}$$

for per-class quantities of (5) indexed by superscript ($i$). Note that (31) becomes exactly (5) for the two-class formulation, when $K = 2$. Again, similarly to the latter case, the particular class to be left out may be determined using domain knowledge, or via statistical techniques, i.e., by maximum within-class variance in the original feature space, etc. In order to accommodate the changes required for adopting (31), the individual matrices $R$ and $G$ from (15) and (24) will be replaced with

$$R_K = \sum_{i=1}^{K-1}\frac{\alpha^{(i)}}{\beta^{(i)}}R^{(i)}, \quad \text{and} \tag{32}$$

$$G_K = \sum_{i=1}^{K-1}G^{(i)}, \tag{33}$$

respectively, where each of the matrices $R^{(i)}$ is computed according to (14) using observations from class $i$, while matrices $G^{(i)}$ are calculated as indicated in (23) with proper index interval adjustment for computing distances between data points of a given class $i$ and the rest of the data set.

## 6 Discussion

In this section we briefly review some of the previously developed approaches of discriminant analysis and dimensionality reduction, demonstrating on simple examples the essential differences between existing techniques and the proposed DDA method.

[6] A word of caution is in order as for the choice of $k = 1$, which corresponds to an ill-posed combinatorial problem [6].

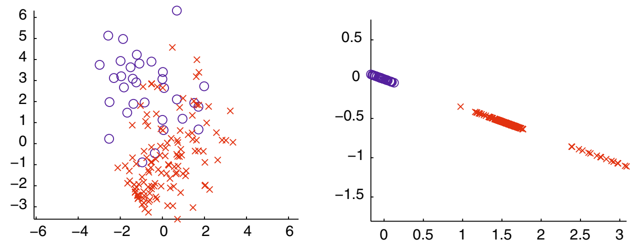
First, we consider principal component analysis (PCA), a fundamental tool for dimensionality reduction that finds a set of orthogonal vectors that account for as much as possible of the data's variance. Apparently, the PCA method disregards class membership information altogether and consequently is of limited use as a discriminatory transform. This conjecture is easily confirmed by comparing 2D projections of the Hepatitis dataset by the PCA and DDA methods illustrated in Fig. 4, which shows a perfect class separation for the latter approach explaining its 100% classification accuracy reported earlier (see Table 2). The singular value decomposition of the resulting transformation reveals that there is only one significantly different from zero singular value, meaning that in order to distinguish between the two classes one may use just one dimension, i.e., project the data set onto a line, as seen in Fig. 4(b).

Fisher's linear discriminant analysis (LDA) [14, 16, 20] projects original data into a smaller number of dimensions, while trying to preserve as much discriminatory information as possible by maximizing the ratio of between-class scatter over within-class scatter. Based on the second order statistical information, the method is proven to be optimal whenever data classes are represented by unimodal Gaussians with well-separated means. A violation of this assumption drastically deteriorates LDA's performance, as seen in Fig. 5 that compares class separation achieved by the projections found by LDA and DDA methods for the classical XOR problem [53]. As for the DDA approach, Fig. 5 illustrates that the proposed technique does not require data Gaussianity assumption. Furthermore, the method can determine discriminative projection transformations of up to as many dimensions as there are in the data, whereas LDA is limited by rank restrictions on the between-class scatter matrices to have no more than $K$–1 dimensions, where $K$ is the number of classes.

A biased discriminant analysis (BDA) approach [62, 63] developed with a goal in mind to improve efficiency of interactive multimedia retrieval applications, is based on an appealing idea of asymmetric treatment of positive and negative relevance feedback examples that is brilliantly conveyed by a famous citation: "All happy families are alike, each unhappy family is unhappy in its own fashion" (L. Tolstoy, *Anna Karenina*). According to this metaphor, the approach seeks a compact representation of the class of positive examples, while the only constraint placed on negative examples is to stay away as far as possible from the positives. This technique excels in overcoming several important drawbacks of LDA induced by scatter matrix rank restrictions and Gaussianity assumptions and, conceptually, is closest to the two-class version of the proposed DDA method. However BDA's implementation is occasionally offset by suboptimal solutions whenever the observations from the two classes overlap considerably along the direction orthogonal to that of minimal variance of the positive examples. An illustration of this adverse condition is depicted in Fig. 6).

Another advantage of relying exclusively on the distances among the observations lets us relax the sought transformation orthogonality condition often found necessary in other methods. For instance, feature transformation based on maximizing mutual information between transformed data and their corresponding class labels proposed by Torkkola et al. [54] parametrizes the transformation via planar rotations and hence is by design orthogonal, as are those of other methods, which operate on orthogonal subspaces.

There also exist other discriminant analysis methods that are specifically designed to work well for non-Gaussian data sets (e.g., NDA [21]) and target the nearest neighbor classifier performance (e.g., a recent enhancement of NDA proposed in [7]), whose main difference from DDA lies in the fact that these methods still rely on parametric within-class scatter matrices. This is likely to explain why these approaches are generally outperformed by the SVM techniques, while DDA demonstrates comparable results (see Table 4).



**(a)** PCA projection **(b)** DDA projection

**Fig. 4** 2D projections of the Hepatitis dataset. **a** PCA projection **b** DDA projection

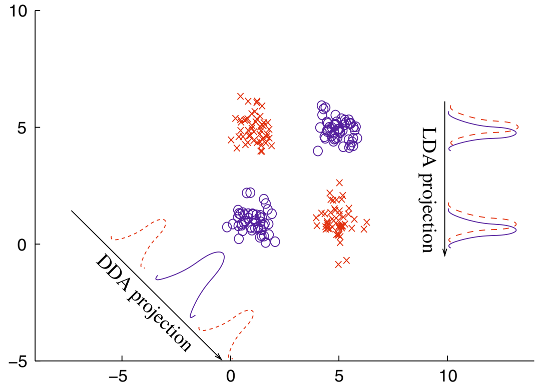


**Fig. 5** XOR problem solution obtained by the LDA and DDA methods

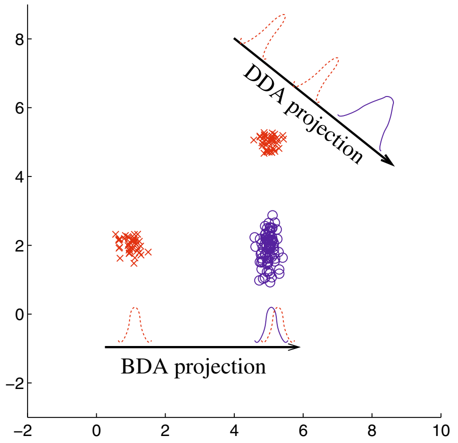
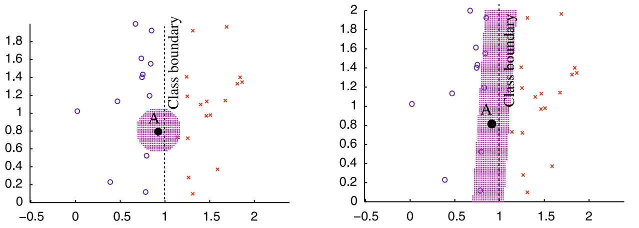


**Fig. 6** Solution of the "dominant variance direction" problem obtained by the BDA and DDA methods

Among iterative techniques, DANN [27] and CDW [45] methods must be highlighted. Similarly to the proposed DDA, the class-dependent weighted (CDW) dissimilarity approach seeks to optimize a certain criterion for improving NN classification accuracy, which is done by deploying the Dinkelbach's algorithm [13] combined with gradient descent. Effectively, a transformation found by the CDW method may be considered a restricted case of the DDA transformation where no dimensionality reduction is allowed and $T$ is required to be diagonal. As opposed to CDW, the discriminant adaptive nearest neighbor (DANN) approach does permit global dimensionality reduction. It operates according to an iterative scheme to obtain a metric modifying local neighborhoods, which makes it different from the DDA in the way that DANN does not optimize any global criterion or objective function. However, both DDA and DANN in many cases lead to similar results, as demonstrated in Fig. 7. This illustration shows how DDA transformation corrects the decision of an NN classifier and, conceptually, is an exact reproduction of the motivational example used by the authors in [27] to describe the intuition behind their technique.

Manifold techniques, such as Isomap [52] and locally-linear embedding (LLE) [47], also present a viable alternative means for dimentionality reduction. However, belonging mostly to the family of unsupervised learning algorithms, they cannot be regarded as directly comparable with the proposed technique that actively uses the class information while deriving the sought discriminative transformation.

In addition to the important differences of the proposed DDA method summarized above, there is yet another advantage to its distance-based formulation which makes it easily applicable for solving more complex non-linear problems via introduction of kernels, as discussed in the section that follows.



**(a)** NN region of **A** (shaded area) in the original space leads to an error

**(b)** NN region of **A** after applying DDA produces a correct classification decision

**Fig. 7** Effect of DDA on local neighborhoods—a comparison to DANN [27]. **a** NN region of A (*shaded area*) in the original space leads to an error **b** NN region of A after applying DDA produces a correct classification decision

## 7 Kernel reformulation of DDA

In this section, we seek to overcome a linearity assumption of the transformation derived by the previously described DDA approach, leading to a formulation of its kernel extension, KDDA. Additionally we focus on a particular aspect of KDDA that opens up a possibility of using indefinite kernels, which stems from a theoretical property of KDDA problem formulation convexity that holds irrespective of the definiteness of the kernel in question.

Suppose there is a space $\mathcal{F}$ where samples of training data can be mapped via $\Phi : \mathbb{R}^m \to \mathcal{F}$, such that there exists a kernel function $k(x,y) = (\Phi(x))^{\mathrm{T}}\ \Phi(y)$, where $x, y \in \mathbb{R}^m$ and $k : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$. We will also assume that the discriminative transformation is sought in $\mathcal{F}$ as a projection matrix $\omega$ of size $[\mathcal{N}_{\mathcal{F}} \times d]$, where $\mathcal{N}_{\mathcal{F}}$ is the dimensionality of $\mathcal{F}$, and $d$ is the dimension of the derived discriminative projection subspace, such that the columns of $\omega$ lie in the span of all training samples mapped in $\mathcal{F}$, by virtue of the Representer Theorem:

$$\omega = \left[\sum_i^N \alpha_i^{(1)}\Phi(z_i)\sum_i^N \alpha_i^{(2)}\Phi(z_i)\cdots\sum_i^N \alpha_i^{(d)}\Phi(z_i)\right], \qquad (34)$$

where $z_i$ is one of the $N_X + N_Y$ samples from the training data compound matrix $Z$, as defined in (25). The distances between images of samples $x$ and $y$ projected from $\mathcal{F}$ by solution $\omega$ are thus expressed as:

$$\begin{aligned}\mathcal{D}^2_{xy}(\omega) &= (\Phi(x) - \Phi(y))^{\mathrm{T}}\omega\omega^{\mathrm{T}}(\Phi(x) - \Phi(y)) \\ &= \mathbf{tr}\left(\omega^{\mathrm{T}}(\Phi(x) - \Phi(y))(\Phi(x) - \Phi(y))^{\mathrm{T}}\omega\right) \\ &= \sum_j^d\left(\sum_i^N \alpha_i^{(j)}(k(z_i, x) - k(z_i, y))\right)^2.\end{aligned} \qquad (35)$$

In matrix notation (35) can be simplified as:

$$\mathcal{D}^2_{xy}(\omega) \equiv \mathcal{D}^2_{xy}(P) = \mathbf{tr}\left(P^{\mathrm{T}}H_{xy}P\right) \qquad (36)$$

where $P \in \mathbb{R}^{N \times d}$ is the sought nonlinear transformation represented as a matrix collecting all of the $\alpha_i^{(j)}$ coefficients, $H_{xy} = (K_x – K_y)(K_x – K_y)^{\mathrm{T}}$, and $K_s = [k(z_1,s), k(z_2,s), \ldots, k(z_N,s)]^{\mathrm{T}}$ denotes a vector of kernel evaluations for sample $s$ over all of the training data.

In view of (36), the logarithm of the DDA optimization criterion (3) can now be expressed in terms of distances projected from a richer, possibly infinite-dimensional feature space $\mathcal{F}$ :

$$\begin{aligned}\log J(P) = &\frac{2}{N_X(N_X-1)} \sum_{i=1}^{N_X} \sum_{j=i+1}^{N_X} \log \Psi\left(\mathscr{D}_{ij}^{W}(P)\right) \\ &- \frac{1}{N_X N_Y} \sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} \log \mathscr{D}_{ij}^{B}(P)\end{aligned} \tag{37}$$

The treatment of the obtained criterion differs only slightly compared to the linear case. Similarly to the way it is done in the DDA method, as described in equations (9)–(26) in Sects. 2 and 3, we express convex parts of the criterion by their respective approximations majorized by quadratics [28], while the concave parts are linearized. The former simple algebraic manipulation relies on the Cauchy-Schwarz inequality, while the latter is a direct consequence of the concavity of the log-function, whose combined application leads to the following approximation:

$$\begin{aligned}\mu_{\log J}(P, \bar{P}) = &\frac{1}{N_X(N_X-1)} \mathbf{tr}\left(P^{\mathrm{T}} \mathbb{K}_X B(\bar{P}) \mathbb{K}_X^{\mathrm{T}} P\right) \\ &+ \frac{1}{2N_X N_Y} \mathbf{tr}\left(P^{\mathrm{T}} \mathbb{K}_{XY} C \mathbb{K}_{XY}^{\mathrm{T}} P\right) \\ &+ \frac{2}{N_X N_Y} \mathbf{tr}\left(P^{\mathrm{T}} \mathbb{K}_{XY} G(\bar{P}) \mathbb{K}_{XY}^{\mathrm{T}} \bar{P}\right) \\ &+ \text{const},\end{aligned} \tag{38}$$

where $\bar{P}$ is the current solution, $\mathbb{K}_X$, $\mathbb{K}_{XY}$ are Gram matrices of kernel inner products evaluated over $X$ and all data, respectively, and $B$, $C$, $G$ are positive semi-definite design matrices independent of $P$ that are derived in a way similar to that shown in "Appendix". Elements $b_{ij}$ of $B$ are defined as:

$$b_{ij} = \begin{cases} -\frac{\bar{w}_{ij}}{\Psi\left(\mathscr{D}_{ij}^{W}(\bar{P})\right)} & \text{if } i \neq j; \\ -\sum_{k=1, k \neq i}^{N_X} b_{ik} & \text{if } i = j; \end{cases} \tag{39}$$

where $\bar{w}_{ij}$ is a weight of the Huber function majorizer, that in this case is equal to 1 if $\Psi(\mathscr{D}_{ij}^{W}(\bar{P}))$ is less than the robustness threshold $c$, or $c / \Psi(\mathscr{D}_{ij}^{W}(\bar{P}))$ otherwise. For matrices $C$ and $G$, their non-zero elements $m_{ij}$ are defined as:

$$m_{ij} = \begin{cases} r_{ij} & \text{for } i \in [1; N_X] \\ & \text{and } j \in [N_X + 1; N], \\ r_{ij} & \text{for } i \in [N_X + 1; N] \\ & \text{and } j \in [1; N_X], \\ -\sum_{k=1, k \neq i}^{N_X + N_Y} m_{ik} & \text{for } i = j, \end{cases} \tag{40}$$

where $r_{ij}$ is equal to –1 and $\frac{-1}{\mathscr{D}_{ij}^{B}(\bar{P})}$ for $C$ and $G$, respectively. Finally, taking into account theoretical considerations mentioned in Sect. 3.3 confirmed by experimental results in Sect. 7.3, we define a regularized formulation

$$\begin{aligned}\mu_{\log J}^{\mathrm{reg}}(P, \bar{P}) = &\frac{1}{N_X(N_X-1)} \mathbf{tr}\left(P^{\mathrm{T}} \mathbb{K}_X B(\bar{P}) \mathbb{K}_X^{\mathrm{T}} P\right) \\ &+ \frac{1}{2N_X N_Y} \mathbf{tr}\left(P^{\mathrm{T}} \mathbb{K}_{XY} C \mathbb{K}_{XY}^{\mathrm{T}} P\right) \\ &+ \frac{2}{N_X N_Y} \mathbf{tr}\left(P^{\mathrm{T}} \mathbb{K}_{XY} G(\bar{P}) \mathbb{K}_{XY}^{\mathrm{T}} \bar{P}\right) \\ &+ \lambda\left(\mathbf{tr}(P^{\mathrm{T}} \mathbb{K}_{XY} P) - \Delta\right),\end{aligned} \tag{41}$$

where a Lagrange multiplier $\lambda$ introduces an $L_2$ norm regularizer expressible as a trace (Representer Theorem).

The approximations used to derive $\mu_{\log J}(P, \bar{P})$ are chosen so as to ensure that the resulting expression's value is never less than the objective to be minimized, and thus provides an upper bound of the criterion (37). By optimizing (38) iteratively, every subsequent iteration achieves a goal function value that is better or at least as good as the one from the previous iteration, which leads to covergence under the practically reasonable objective boundedness assumption.

More formally, such an iterative scheme that constitutes the core of the KDDA, the kernelized extension of the distance-based discriminant analysis method, can be written as the following algorithm:

**Algorithm KDDA**

```
1. Assign an initial starting point $\bar{P} = \bar{P}_0 \in \mathbb{R}^{N \times d}$, set
   convergence tolerance $\varepsilon$;
2. Find a successor point $P_s : P_s = \arg\min_P \mu_{\log J}(P, \bar{P})$
   subject to a regularization constraint;
3. If $\log J(\bar{P}) - \log J(P_s) < \epsilon$, then stop;
4. Set $\bar{P} = P_s$, go to 2.
```

## 7.1 Indefinite kernels via hyperkernels

In contrast to the vast majority of kernel-based techniques for discriminant analysis and classification whose numerical stability, convergence and theoretical performance guarantees depend crucially on the positive semi-definiteness (PSD) of the underlying kernel function, the KDDA method is free from such a restriction. Indeed, the computationally convenient convexity of the described above approximation (38) is due to the PSD property of matrices $B$ and $C$ only, which is true by construction (see "Appendix"), and hence is not affected even when the so-called *indefinite kernels* [23, 44] are applied. These kernels do not satisfy Mercer's theorem in the strict sense and hence may produce indefinite Gram matrices, presenting some difficulties to the traditional computational methods [23]. Nevertheless, an impressive suite of indefinite kernel methods have been proposed and proven effective in practice by successfully applying jittered [11], tangent distance [24], Kullback-Leibler divergence [41], dynamic time warping [2], distance substitution [25] indefinite kernel functions. In addition to these empirical results, there exist some important theoretical contributions and facts on indefinite kernels as well, such as the recent studies on Reproducing Kernel Krein Spaces (RKKS) [44], the indefiniteness of the sigmoid kernel $k(x,x') = \tanh(ax^T x' + b)$ of neural networks for certain paramter range [36, 57], or convenient convex SVM problem formulations obtained with a broad class of conditionally positive definite kernels [48], the geometric margin interpretation attainable for indefinite kernels producing co-oriented projected and feature space separating hyperplane normal vectors [23], as well as many other results and efforts that motivate further examination of indefinite kernels in the KDDA framework, especially given the fact that KDDA by design is built to tolerate indefinite kernels. In the discussion that follows we consider the application of the hyperkernel method [43] within the KDDA framework with an important modification—the removal of the kernel PSD constraint.

### 7.2 Overview of hyperkernel method

The approach of hyperkernels [43] automatically adjusts kernel parameters in a data-dependent fashion and uses the kernel trick on the space of kernels in order to be able to control the complexity of the learned kernel function via a regularized quality functional $Q_{\text{reg}}$. By analogy with the definition of the regularized risk functional $R_{\text{reg}}$ commonly used in the support vector machines [10, 58]:

$$R_{\text{reg}} = R_{\text{emp}} + \lambda ||f||^2_{\mathcal{H}} \tag{42}$$

the regularized quality functional $Q_{\text{reg}}$ is a sum of a quality functional $Q_{\text{emp}}$ and a regularization term:

$$Q_{\text{reg}} = Q_{\text{emp}} + \lambda_Q ||k||^2_{\underline{\mathcal{H}}} \tag{43}$$

where the former term tells how well matched kernel $k$ is to the given data set, while the latter is the norm of the kernel in Hyper-RKHS $\underline{\mathcal{H}}$ for some positive regularization constant $\lambda_Q$. The insight of the hyperkernel approach that specifies $\underline{\mathcal{H}}$ and finds an appropriate kernel in an infinite space of possible solutions much in the same way a suitable hypothesis is found in the RKHS induced by a fixed kernel in the regularized risk minimization problem, is based on an appealing and elegant idea. Namely, the method defines a compound set $\underline{\mathcal{X}} = \mathcal{X} \times \mathcal{X}$ treating kernel $k$ as a function $k : \underline{\mathcal{X}} \to \mathbb{R}$, which allows to extend the definition of an RKHS for the case of a hyperkernel $\underline{k} : \underline{\mathcal{X}} \times \underline{\mathcal{X}} \to \mathbb{R}$, thus arriving at the concept of Hyper-RKHS, $\underline{\mathcal{H}}$. More importantly, it is shown that the Representer Theorem holds for Hyper-RKHS. In other words, even though the optimization of $Q_{\text{reg}}$ may be carried over a whole space of kernels, it is still possible to find an optimal solution of (43) by choosing among a finite number.

### 7.3 Indefinite KDDA

Note that the kernel obtained as a free linear combination of hyperkernels is not necessarily positive semidefinite [37], which is why the original hyperkernel method imposes an additional constraint and ends up solving a semidefinite optimization problem when $Q_{\text{emp}}$ is replaced with a standard formulation of regularized risk functional (42). However, in the case of KDDA, we are not restricted by this PSD requirement and by virtue of the Representer Theorem for Hyper-RKHS can replace $Q_{\text{emp}}$ with (41). Furthermore, the co-orientation condition [23] is automatically fulfilled by the regularization term of the KDDA formulation. Thus, the regularized quality functional minimization problem in the KDDA case becomes:

$$\begin{aligned} Q^{\text{KDDA}}_{\text{reg}} &= \mu_{\log J}(P, \bar{P}, \beta, \bar{\beta}) \\ &\quad + \lambda\left(\mathbf{tr}(P^{\mathrm{T}} K(\beta) P) - \Delta\right) \\ &\quad + \lambda_Q \beta^{\mathrm{T}} \underline{K} \beta \end{aligned} \tag{44}$$

where the approximation of the criterion sought to be minimized $\mu_{\log J}(P, \bar{P}, \beta, \bar{\beta})$ now depends on hyperkernel expansion coefficients $\beta_{i,j}$ collected in vector $\beta$ in addition to $P, \underline{K}$ is a hyperkernel Gram matrix, $K(\beta)$ is a $N \times N$ kernel matrix obtained by reshaping an $N^2$-element vector $\underline{K}\beta$, and $\lambda$ and $\Delta$ are regularization parameters. Finally, a practical solution scheme is obtained by breaking down (44) into a two-stage alternating optimization problem with a projection stage, that solves (44) for $P$ while fixing current $\beta$, and a hyperkernel stage, that solves (44) for $\beta$ while fixing current $P$. In summary, the iterative procedure of the KDDA method with indefinite kernels can be stated as follows:

**Algorithm Indefinite KDDA.**

```
1. Assign an initial starting point $\bar{P} = \bar{P}_0 \in \mathbb{R}^{N \times d}$, $\bar{\beta} = \bar{\beta}_0 \in \mathbb{R}^{N^2}$, set tolerance $\varepsilon$
2. Fix $\beta$ and solve projection stage:
       $P = \arg\min_{P} \mu_{\log J}(P, \bar{P})$
3. Fix $P$ and solve hyperkernel stage:
       $\beta = \arg\min_{\beta} \mu_{\log J}(\beta, \bar{\beta})$
4. If $\log J(\bar{P}, \bar{\beta}) - \log J(P, \beta) < \epsilon$, then stop
5. Set $\bar{P} = P, \bar{\beta} = \beta$ and go to 2
```

Notably, step 2 of the above algorithm involves the same optimization formulation as the one detailed in the previous section 6, provided that the new Gram matrices have been recomputed and fixed, such that $\mathbb{K}_{XY} \equiv K(\beta)$. The problem from step 3 essentially reduces to a large-scale convex quadratic minimization problem with a single linear constraint, instead of the original hyperkernel method's SDP problem solving which, in general, takes longer than solving a quadratic program [43]. Similarly to the other variants of the algorithm discussed before, the iterative procedure for indefinite KDDA converges because of the boundedness of the objective function and stage-wise improvement at each iteration.

## 8 Experimental results

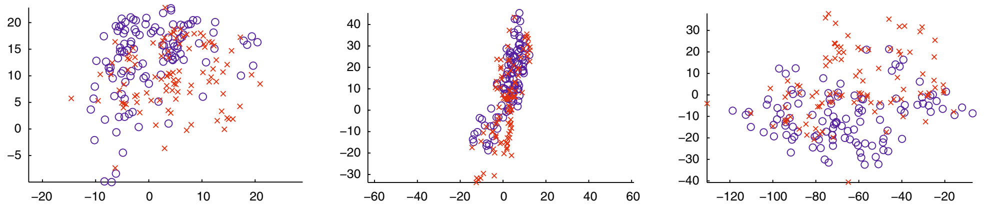
### 8.1 UCI Benchmark data set performance

Our preliminary empirical analysis was based on data sets from the UCI Machine Learning Repository [5]. First of all, we verified that the solutions of the optimization problem formulated in Sect. 2 found by the proposed method were of better quality and less dependent on the choice of the initial value compared to those of generic techniques, confirming the results reported by Van Deun [56] and Webb [60]. Indeed, numerous random initializations of the gradient descent, together with its stochastic variant, led to inferior as well as unstable results reflected in higher values of log $J$ (see examples of 2D discriminant projection of Sonar data set in Fig. 8), while the IM-based method regularly reached far lower criterion values, as seen in Fig. 9, and proved nearly insensitive to the choice of the initial supporting point. In addition to that, we thoroughly verified that the convergence property of the IM procedure was indeed preserved, as illustrated in Fig. 9, despite the use of a Taylor series approximation in the derivation of (26). Finally, we validated the proposed dimensionality reduction technique by analysing how the classification performance varied with respect to $k$, the dimensionality of the transformed space, and how it was related to the number of non-zero singular values of the full-dimensional transformation, an example of which for the Sonar data set is depicted in Fig. 10.

Figure 10b plots 10 largest singular values of the full-dimensional transformation, in descending order, while Fig. 10 documents the results of 10-fold cross-validation performance with respect to the transformed space dimensionality. It is easy to see that the singular values beyond the seventh one are virtually zero, which corresponds to the point after which increasing the transformed space dimensionality, by either setting $k$ to a particular value (dot-filled bars) or using a larger number of appropriately scaled left-singular vectors (shaded bars), no longer significantly improves the classification performance, as confirmed by Chow test for structural change [8] at 99% confidence.

Further, the results of classification performance in terms of error rate of two types of experiments were compared. For the first type of experiments, which we will refer to as simply "NN" experiments, we measured classification error rate of the NN classifier using 10-fold cross-validation [61]. In the second type of experiments, that are going to be called "DDA+NN" experiments, an additional stage of applying a discriminating transformation $T$ derived with the proposed DDA method prior to measuring the cross-validation performance of the NN classifier was introduced. Therefore, the goal of this analysis was to assess the effect of applying a DDA transformation on the accuracy of the NN classifier.

Several well-known data sets from the UCI Machine Learning Repository [5] were used in our experiments. All of the available data from each data set were utilized on the "as is" basis without performing any preprocessing, such



**(a)** $\log J = -0.17$ **(b)** $\log J = -0.22$ **(c)** $\log J = -0.19$

**Fig. 8** Sonar data: local minima-prone solutions found by the gradient descent method. The target dimensionality of the sought discriminative subspace was set to $k = 2$. **a** log $J = -0.17$, **b** log $J = -0.22$, **c** log $J = -0.19$

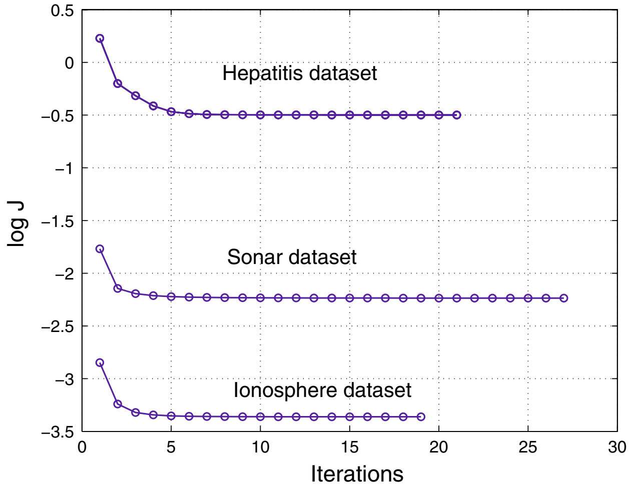
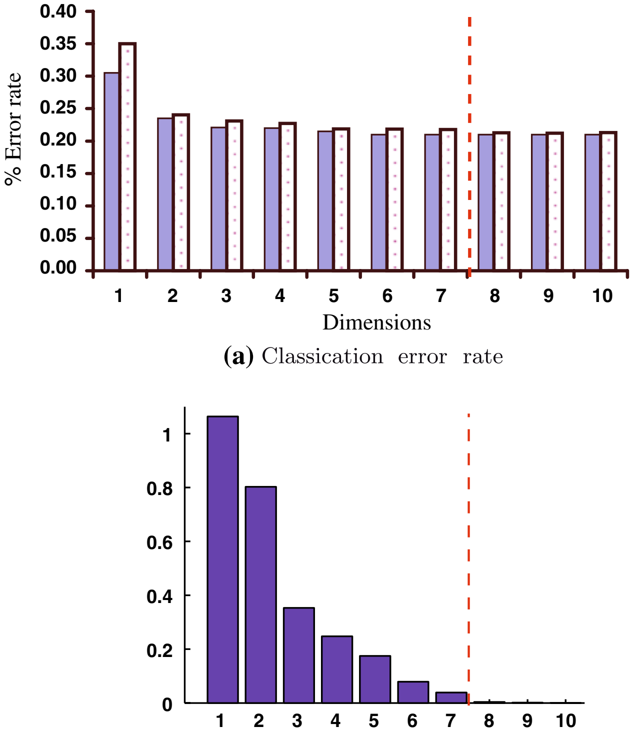


**Fig. 9** Convergence of the iterative majorization procedure in the DDA method. The horizontal and vertical axes correspond to the iteration number and optimization criterion value, respectively



**(a)** Classication error rate

**(b)** Singular values

**Fig. 10** Dimensionality reduction experiments: classification performance results and singular values of the transformation matrix. The *dashed lines* mark the boundary that determines the sufficient dimensionality of the transformed space. **a** Classication error rate **b** Singular values

**Table 1** Summary of data set characteristics

| Data set | Classes | Attributes | Examples |
|---|---|---|---|
| Hepatitis | 2 | 19 | 155 |
| Ionosphere | 2 | 34 | 200 |
| Diabetes | 2 | 8 | 768 |
| Heart | 2 | 13 | 270 |
| Monk's Problem 1 | 2 | 6 | 432 |
| Balance | 3 | 4 | 625 |
| Iris | 3 | 4 | 150 |
| DNA | 3 | 180 | 2000 |
| Vehicle | 4 | 18 | 846 |

as feature expansion for categorical, discrete or binary attributes. For some datasets, specific instructions were supplied as for partitioning the data into the training and testing portions, in which cases cross-validation procedure was not applied. The summary of important characteristics of the data sets used for testing is shown in Table 1. The error rates of NN and DDA+NN data classification experiments averaged over twenty trial cross-validation runs are presented in Table 2. The obtained results confirm our conjecture about the positive effect of applying the DDA transformation on the accuracy of the NN classifier showing an improvement in performance (see Table 2).

## 8.2 Low-level feature representation

In order to assess the proposed DDA method in the context of the semantic augmentation domain, we perform a number of basic experiments of visual object recognition, categorization and semantic retrieval, where multimedia data is provided in the form of digital images and an algorithm is examined to determine how well it can learn the associated semantic information. Before detailing these experiments, however, we take a closer look at the low-level visual feature representation of the said image data, as extracted by the *Viper* system [51].

*Viper* uses a palette of 166 colors, derived by uniformly quantizing the cylindrical *HSV* color space into 18 hues, 3 saturations, and 3 values. These are augmented by four gray levels. This choice of quantization means that more tolerance is given to changes in saturation and value, which is desirable since these channels can be affected by lighting conditions and viewpoint. The choice of the *HSV* color space is due to its perceptual uniformity and a relatively low complexity of computation and inversion in comparison to such alternatives as *CIE-LUV* and *CIE-LAB* [50].

As for the texture features, *Viper* employs a bank of real, circularly symmetric Gabor filters, proposed by Fogel and Sagi [19] and used successfully in image processing applications for image retrieval [26], texture segmentation [15] and face recognition [49]. These filters are defined in the spatial domain as follows:

**Table 2** Classification results for UCI data sets

| Data set | % Error of NN | % Error of DDA+NN |
|---|---|---|
| Hepatitis | 29.57 | 0.00 |
| Ionosphere | 13.56 | 7.14 |
| Diabetes | 30.39 | 27.11 |
| Heart | 40.74 | 21.11 |
| Monk's P1 | 14.58 | 0.69 |
| Balance | 21.45 | 3.06 |
| Iris | 4.00 | 3.33 |
| DNA | 23.86 | 6.07 |
| Vehicle | 35.58 | 24.70 |

$$f_{mn}(x,y) = \frac{e^{-\frac{x^2+y^2}{2\sigma_m^2}}}{2\pi\sigma_m^2}\cos[2\pi(u_{0_m}x\cos\theta_n + u_{0_m}y\sin\theta_n)], \qquad (45)$$

where $m$ indexes the scales of the filters, and $n$ their orientations. The center frequency of the filter is specified by $u_{0\_m}$. The half-peak radial bandwidth is given by:

$$B_r = \log_2\left(\frac{2\pi\sigma_m u_{0_m} + \sqrt{2\ln 2}}{2\pi\sigma_m u_{0_m} - \sqrt{2\ln 2}}\right), \qquad (46)$$

where $B_r$ is chosen to be 1, i.e. a bandwidth of one octave, which then allows us to compute $\sigma_m$:

$$\sigma_m = \frac{3\sqrt{2\ln 2}}{2\pi u_{0_m}}. \qquad (47)$$

The highest center frequency is $u_{0_1} = \frac{0.5}{1+\tan(1/3)} \approx 0.5$, so that it is within the discrete frequency domain. The center frequency is halved at each change of scale, which implies that $\sigma$ is doubled (47). The orientation of the filters varies in steps of $\pi/4$, and three scales are used. These choices result in a bank of 12 filters, which renders appropriate coverage of the frequency domain with little overlap between the filters. Given the 10 band energy quantization, this design provides 120 global texture characteristics of the image. Combining this information with the color data, we obtain a common 286-dimensional feature vector representation for every image.

### 8.3 Application to visual object recognition

For our object recognition experiments we chose a recently developed database ETHZ80 for object categorization and recognition composed of entities corresponding to the basic level of human knowledge organization [34]. The database contains high-resolution color images of 80 objects from eight different classes, for a total of 3,280 images, an overview of which is shown in Fig. 11.

The training set comprised images taken one per class object viewed from a fixed position, while the rest (3,200 images) was allocated to the test set. An illustration of a training set image from class "car" and several test set images is provided in Fig. 12. Again, similarly to the setup described above (see Sect. 7.1), we compared performance results for "NN"and "DDA+NN" experiments for each of the eight classes, but this time, using a one-against-all classification configuration typically encountered in ensemble learning [12], and setting target dimensionality to 2D according to the magnitude of the transformation singular values as explained in Sect. 4.2. The results are summarized in Table 3.

It is importnant to emphasize here that image representation for these experiments was reduced via DDA to two dimensions only. Nevertheless, as shown in the last column of Table 3, the proposed technique still was able to descrease recognition error rate, which improved the overall performance average. The results in Table 3 also reveal the importance of the length constraint (or, regularization), introduced in (27), for the purpose of avoiding data over-fitting problems. Both unconstrained and length-constrained solutions found by the DDA procedure lead to zero error rate on the training data, but, as can be easily seen from Table 3, their performance turned out to be drastically different on the test data sets, demonstrating an adequate generalization capability induced by the length-constrained version of the proposed method. Consistent with the figures reported earlier for color- and texture-based feature sets [34], the error rates are highest for classes 3, 5 and 6. An example of the 2D representation of the training set for image class 2 obtained by DDA is shown in Fig. 13. As can be easily seen from the figure, the target class images are well separated from those of all of

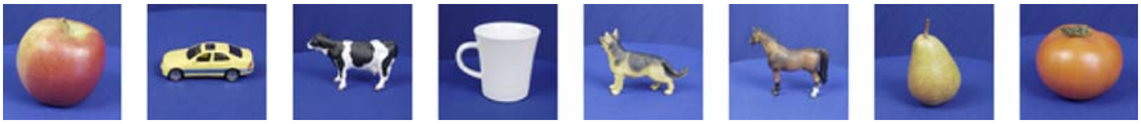


**Fig. 11** The eight classes of objects of the ETHZ-80 database. Each class contains 10 objects with 41 views per object, for a total of 3,280 images

**Fig. 12** An illustration of images of the same class used in the training (leftmost) and test (the rest) sets

**Table 3** Object recognition results for the ETHZ80 image database

| Object class | % Error of NN | % Error of DDA+NN (unconstrained) | %Error of DDA+NN (constrained) |
|---|---|---|---|
| Apple | 4.47 | 18.66 | 0.75 |
| Car | 14.47 | 18.72 | 5.78 |
| Cow | 12.12 | 16.91 | 10.97 |
| Cup | 3.09 | 16.94 | 2.22 |
| Dog | 14.00 | 16.66 | 12.72 |
| Horse | 14.47 | 14.84 | 13.16 |
| Pear | 6.13 | 18.94 | 3.84 |
| Tomato | 2.50 | 16.87 | 1.88 |

the other classes seen to be freely mixed together in the derived 2D discriminative subspace, which is exactly the requirement one seeks to satisfy in one-against-all classification. Additionally, the separation margin visually noticeable in the shown projection suggests that the proposed method may perform as well or better as margin-based techniques.

### 8.4 Application to semantic image retrieval

In addition to the tests mentioned above, we also explored empirically the influence of the DDA transformation on the performance of other classification methods, including NN as a baseline, on the task of semantic image retrieval. For these experiments, three potentially overlapping image sets were selected from the Washington University annotated image collection [35], based on the presence of keywords "trees", "cars" and "ocean" in their annotation. Every classifier was then tested by 10-fold cross-validation. The results of these experiments demonstrate that applying the DDA transformation not only consistently improves NN classifier accuracy, but also provides a boost in performance to some more advanced non-linear classification methods, such as SVM [10], as shown in Table 4.

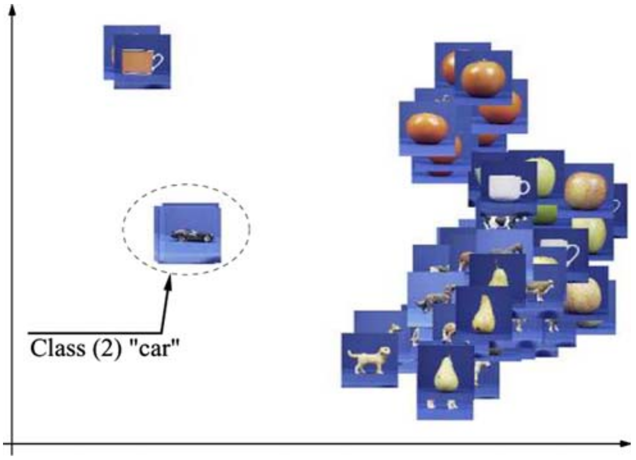


**Fig. 13** Result of applying a discriminative dimensionality-reducing (286 to 2) DDA transformation to the training set for recognition of objects from class (2) "car". Images from class 2 are projected close to each other while images belonging to the other classes are freely scattered maintaining a certain distance margin from class 2

The latter finding emphasises the importance of the alternative interpretation we gave to the DDA method in Sect. 4.2. That is, in addition to the explicitly sought transformation $T$, the technique may also be seen as providing a discriminative distance metric $TT^{\mathrm{T}}$ that accounts for differences in the scales of different features, removes global correlations and redundancies among features to some extent, and adapts to the fact that some features may be much more informative about the class labels than others. This observation is easily illustrated by the example of SVM classifier with a Gaussian kernel:

$$k_{\Sigma}(x_i, x_j) = \mathrm{e}^{-(x_i-x_j)^{\mathrm{T}}\Sigma^{-1}(x_i-x_j)}, \qquad (48)$$

for some covariance matrix $\Sigma$ and observations $x_i$, $x_j$ represented as column vectors. A typical choice of $\Sigma$ here is an identity matrix multiplied by some constant factor. However, when the DDA technique is applied to preprocess the training data before the SVM learning occurs, the SVM classifier fully takes advantage of the discriminative features extracted by the DDA method since

**Table 4** Semantic image retrieval results

| Classifier | % Error on image data set | | |
|---|---|---|---|
| | Trees | Ocean | Cars |
| Fisher's LDA | 43.89 | 45.56 | 17.72 |
| SVM (linear) | 31.11 | 21.11 | 1.58 |
| DDA+SVM (linear) | 17.78 | 11.11 | 1.40 |
| SVM (gaussian) | 23.89 | 16.67 | 1.58 |
| DDA+SVM (gaussian) | 17.78 | 11.11 | 1.40 |
| NN | 38.33 | 19.44 | 2.46 |
| DDA+NN | 18.89 | 18.33 | 1.23 |

the kernel products can now be seen as evaluated in a new discriminative metric $TT^{\mathrm{T}}$:

$$k_{\Sigma}(x_i, x_j) = \mathrm{e}^{-(x_i-x_j)^{\mathrm{T}} TT^{\mathrm{T}} (x_i-x_j)}. \tag{49}$$

This eventually results in SVM being able to find a simpler solution involving fewer support vectors and better generalization properties, which naturally leads to an improvement in classification performance, as shown in Table 4.

From the empirical point of view, in order to verify that non-trivial collection-independent learning has occurred, we also examined the categorization performance of the derived above category-specific DDA transformations on a completely separate image set taken from the COREL database. The empirical evidence demonstrates that the application of the DDA transformation leads to robust categorization of unseen images producing semantically relevant matches that may (Fig. 14, row one) or may not (Fig. 14, row two) share the same vocabulary with the query category, as well as allowing images to be assigned to multiple relevant categories (Fig. 14, the last two images in both rows).

### 8.5 Evaluation of kernel-based extensions

As a basis for comparison with the proposed method of *Indefinite KDDA*, Sect. 6.3, we used related discriminant analysis techniques, already mentioned in the previous sections: Kernel Fisher Discriminant (KFD), Kernel Biased Discriminant Analysis (BiasMap), and KDDA with a fixed kernel function. Kernel parameters for these approaches were determined by cross-validation, and fixed throughout. The parameters for the Indefinite KDDA technique were set to $\Delta = 1$ by using a validation data set, while hyper-kernel parameters were specified as $\lambda_h = 0.6$ to provide an adequate coverage of various kernel widths by the Gaussian harmonic hyperkernel and $\lambda_Q = 1$ according to the recommendations from the authors of the hyperkernel approach [43]. The obtained results for each method in terms of geometric mean accuracy evaluated on the ETHZ80 digital image collection are given in Table 5. Here, we see that the indefinite kernel extension of the KDDA technique enhances the baseline KDDA method fine-tuned by cross-validation with a resulting increase of accuracy from 76.78 to 83.06%. In addition to that, one may observe that the proposed approach outperforms, albeit sometimes by a small margin, all other alternative discriminant analysis techniques considered. It also should be noted that in all eight semantic category classes, the spectra of the Gram matrices at convergence contained

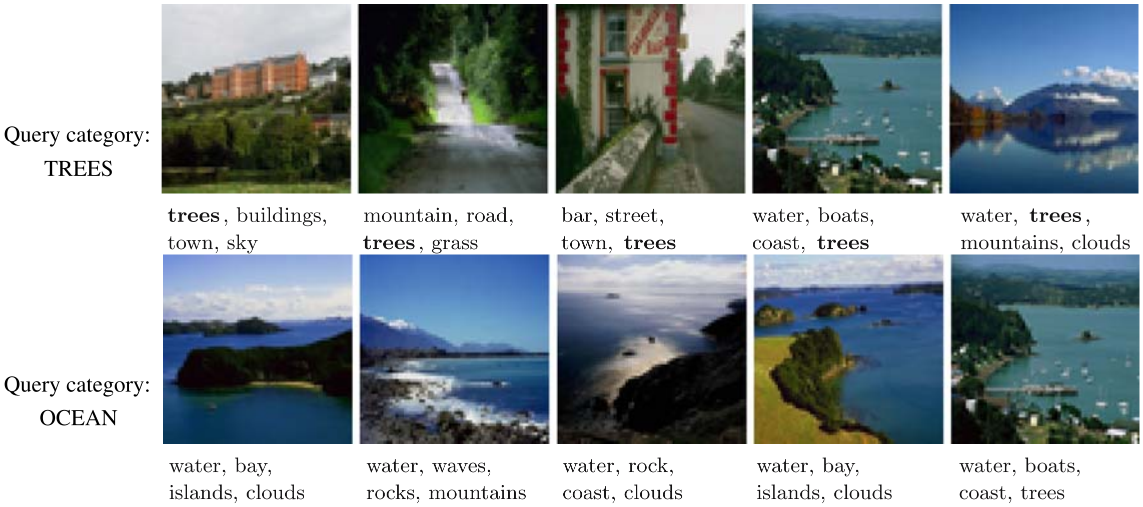

Query category: TREES
**trees**, buildings, town, sky
mountain, road, **trees**, grass
bar, street, town, **trees**
water, boats, coast, **trees**
water, **trees**, mountains, clouds
Query category: OCEAN
water, bay, islands, clouds
water, waves, rocks, mountains
water, rock, coast, clouds
water, bay, islands, clouds
water, boats, coast, trees

**Fig. 14** Examples of semantic image retrieval. The semantic query specified as a natural language keyword is shown on the left. The true (manually assigned) annotation keywords are listed underneath each image. The annotation keywords overlapping with the query are in *bold* font

**Table 5** Object categorization results for the ETHZ80 image database in terms of geometric mean accuracy (in %)

| Object class | KFD | BiasMap | KDDA | Indef. KDDA |
|---|---|---|---|---|
| Apple | 90.35 | 61.56 | 86.02 | 83.21 |
| Car | 76.62 | 72.27 | 66.39 | 82.86 |
| Cow | 59.02 | 53.40 | 56.51 | 69.25 |
| Cup | 94.69 | 56.37 | 87.06 | 93.49 |
| Dog | 76.09 | 40.09 | 70.86 | 78.31 |
| Horse | 81.25 | 39.06 | 67.00 | 76.95 |
| Pear | 86.76 | 68.73 | 86.91 | 86.39 |
| Tomato | 96.66 | 72.73 | 93.45 | 94.05 |
| Average | 82.68 | 58.03 | 76.78 | 83.06 |

both negative and positive eigenvalues, thus confirming the hypothesis on the usefulness of indefinite kernels.

## 9 Concluding remarks

We have described a formulation, extensions and applications of a non-parametric distance-based discriminant analysis technique. The presented method focuses on finding a transformation of the original data that enhances its degree of conformance to the compactness hypothesis and its inverse, which has been shown to lead to a better recognition performance. The classification accuracy has been demonstrated to improve when combined with popular classifiers such as NN and SVM. The latter result underlines the important alternative use of the derived transformation in the capacity of a discriminative metric that accounts for differences in the scales of different features, removes to some extent global correlations and redundancies, and adapts to the fact that some features may be much more informative about the class labels than others.

The presented DDA formulation extends naturally from binary to multiple class discriminant analysis problems. The method can also serve as a discriminating dimensionality reduction technique with the ability to overcome the limitation of the classical parametric approaches that typically extract at most $K$–1 features for a $K$-class problem, while possessing the means to determine in a data-dependent fashion how many dimensions are sufficient to distinguish among a given set of classes. Also considered are the possible extensions of the proposed approach to more complex non-linear problems via kernels, as well as applicability of the technique for the case of non-positive definite kernels.

We have verified the classification performance of the proposed method and its extensions on a number of the benchmark data sets from UCI Machine Learning Repository [5] and on the real-world semantic image retrieval tasks. The encouraging results demonstrated that the method outperforms several popular methods, and improves classification accuracy, sometimes dramatically, making it an excellent candidate for a number of application in pattern recognition, classification, categorization domains.

## 10 Appendix

This section focuses on the intuition behind the definitions of design matrices $R$ and $G$ specified in (14) and (22). The derivations listed here are mostly based on those developed for the SMACOF multi-dimensional scaling algorithm [6].

Let us consider matrix $R$ that is used in calculation of the majorizing expression of $S_W(T)$ represented by a weighted sum of within-distances. In the derivations that follow, we will assume all weights to be equal to unity, and show afterwards how this assumption can be easily corrected for. We, thus, begin by rewriting a squared within-distance in the vector form:

$$\left(d_{ij}^W(T)\right)^2 = \sum_{a=1}^{m}(x'_{ia} - x'_{ja})^2 = (\mathbf{x}'_i - \mathbf{x}'_j)(\mathbf{x}'_i - \mathbf{x}'_j)^{\mathrm{T}}, \qquad (50)$$

where $\mathbf{x}'_i$ and $\mathbf{x}'_j$ denote rows $i$ and $j$ from matrix $X' = XT$, representing the corresponding observations transformed by $T$. Noticing that $\mathbf{x}'_i - \mathbf{x}'_j = (e_i–e_j)^{\mathrm{T}} X'$, (50) becomes:

$$\begin{aligned}\left(d_{ij}^W(T)\right)^2 &= (e_i - e_j)^{\mathrm{T}} X' X'^{\mathrm{T}} (e_i - e_j)\\ &= \mathbf{tr}\left(X'^{\mathrm{T}}(e_i - e_j)(e_i - e_j)^{\mathrm{T}} X'\right)\\ &= \mathbf{tr}\left(X'^{\mathrm{T}} A_{ij} X'\right),\end{aligned} \qquad (51)$$

where $A_{ij}$ is a square symmetric matrix whose elements are all zeros, except for those four indexed by the combinations of $i$ and $j$ that are either 1 (diagonal) or –1 (off-diagonal). For instance, $A_{13}$ for $i = 1, j = 3$ and $N_X = 3$ will have the following form:

$$A_{13} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}. \qquad (52)$$

Taking into account (51), the sum of the squared within-distances can be expressed as:

$$\begin{aligned}\sum_{i<j}^{N_X}\left(d_{ij}^W(T)\right)^2 &= \sum_{i<j}^{N_X}\mathbf{tr}\left(X'^{\mathrm{T}} A_{ij} X'\right)\\ &= \mathbf{tr}\left(X'^{\mathrm{T}} V X'\right)\\ &= \mathbf{tr}\left(T^{\mathrm{T}} X^{\mathrm{T}} V X T\right),\end{aligned} \qquad (53)$$

where $V = \sum_{i<j}^{N_X} A_{ij}$, for which there exists an easy computational shortcut. Namely, $V$ is obtained by placing

–1 in all off-diagonal entries of the matrix, while the diagonal elements are calculated as negated sums of their corresponding off-diagonal values in rows or columns. That is:

$$v_{ij} = \begin{cases} -1, & \text{if } i \neq j; \\ -\sum\limits_{k=1,k\neq i}^{N_X} v_{ik} = N_X - 1, & \text{if } i = j; \end{cases} \tag{54}$$

For instance, coming back to our previous $N_X = 3$ example, this technique produces:

$$\begin{aligned} V &= \sum_{i<j}^{N_X=3} A_{ij} \\ &= \left( \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix} \right) \\ &= \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}. \end{aligned} \tag{55}$$

It is not difficult to see that the same result applies to the case of non-unitary weights associated with each distance, the only difference being that instead of –1 placed into the off-diagonal elements of $V$, one should use the negated values of the corresponding weights. And this is exactly how the matrix formulation of $\mu_{S_W}(T, \bar{T})$, (15), and design matrix $R$, (14), are obtained:

$$\begin{aligned} \mu_{S_W}(T, \bar{T}) &= \sum_{i<j}^{N_X} \frac{\bar{w}_{ij} \cdot \left(d_{ij}^W(T)\right)^2}{2\Psi\left(d_{ij}^W(\bar{T})\right)} + K_1 \\ &= \sum_{i<j}^{N_X} \frac{\bar{w}_{ij}}{\Psi\left(d_{ij}^W(\bar{T})\right)} \left[\frac{1}{2}\mathbf{tr}\left(T^{\mathrm{T}}X^{\mathrm{T}}A_{ij}XT\right) + K_1'\right] \\ &= \frac{1}{2}\mathbf{tr}\left(T^{\mathrm{T}}X^{\mathrm{T}} \sum_{i<j}^{N_X} \frac{\bar{w}_{ij}}{\Psi\left(d_{ij}^W(\bar{T})\right)} A_{ij}XT\right) + K_1 \\ &= \frac{1}{2}\mathbf{tr}\left(T^{\mathrm{T}}X^{\mathrm{T}}RXT\right) + K_1 \end{aligned} \tag{56}$$

In order to derive the formulation of matrix $G$, as specified for the majorizer of $-S_B(T)$ based on Taylor series expansion (23), we rewrite (22) using the same techniques as we did in (51) arriving at:

$$-d_{ij}^B(T) \leq -\frac{\mathbf{tr}\left(T^{\mathrm{T}}Z^{\mathrm{T}}C_{ij}Z\bar{T}\right)}{d_{ij}^B(\bar{T})}, \tag{57}$$

where $C_{ij} = (e_i – e_{N\_X+j})\ (e_i – e_{N\_X+j})^{\mathrm{T}}$ is a between-class analog of matrix $A_{ij}$. From (55), it is apparent that the same type of a computational shortcut used above to obtain $V$ may be exploited here too. Indeed, matrix $F = \sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} C_{ij}$ can be quickly constructed by placing –1 in the off-diagonal elements that correspond to index locations of the between-distances, and subsequently summing with negation to obtain the diagonal entries. An illustration of the technique for $N_X = 2$, $N_Y = 3$ is shown below:

$$\begin{aligned} F &= \sum_{i=1}^{N_X=2} \sum_{j=1}^{N_Y=3} C_{ij} \\ &= \begin{bmatrix} 3 & 0 & -1 & -1 & -1 \\ 0 & 3 & -1 & -1 & -1 \\ -1 & -1 & 2 & 0 & 0 \\ -1 & -1 & 0 & 2 & 0 \\ -1 & -1 & 0 & 0 & 2 \end{bmatrix}. \end{aligned} \tag{58}$$

This is the case of unitary weights. Again, the extension to the non-unitary weight formulation is trivial, and will involve pre-multiplying the off-diagonal entries by the appropriate quantities, which in the case of $G$ are the reciprocals of the squares of the corresponding distances, as shown in (23).

## References

1. Arkadev A, Braverman E (1966) Computers and pattern recognition. Thompson, Washington, DC
2. Bahlmann C, Haasdonk B, Burkhardt H (2002) On-line handwriting recognition with support vector machines—a kernel approach. In: Eighth International Workshop on Frontiers in Handwriting Recognition. Ontario, Canada
3. Bartlett P (1997) For valid generalization, the size of the weights is more important than the size of the network. Adv Neural Inform Process Syst 9:134–140
4. Bertero M, Boccacci P (1998) Introduction to inverse problems in imaging. Institute of Physics Publishing
5. Blake CL, Merz CJ (1998) UCI repository of machine learning databases
6. Borg I, Groenen PJF (1997) Modern multidimensional scaling. Springer, New York
7. Marco Bressan, Vitria J (2003) Nonparametric discriminant analysis and nearest neighbor classification. Pattern Recogn Lett 24(15):2743–2749
8. Chow GC (1960) Tests of equality between sets of coefficients in two linear regressions. Econometrica 28(3)
9. Commandeur J, Groenen PJF, Meulman J (1999) A distance-based variety of non-linear multivariate data analysis, including weights for objects and variables. Psychometrika 64(2):169–186
10. Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge
11. Dennis DeCoste, Bernhard Schölkopf (2002) Training invariant support vector machines. Mach Learn 46(1–3):161–190
12. Dietterich TG (2000) Ensemble methods in machine learning. In: Kittler J, Roli F (eds) First international workshop on multiple classifier systems. Springer, Heidelberg, pp 1–15

13. Dinkelbach W (1967) On nonlinear fractional programming. Manage Sci A(13):492–498
14. Duda RO, Hart PE (1973) Pattern classification and scene analysis. Wiley, New York
15. Dunn D, Higgins WE, Wakeley J (1994) Texture segmentation using 2-d gabor elementary functions. IEEE Trans Pattern Anal Mach Intell 16(2):130–149
16. Fisher RA (1936) The use of multiple measures in taxonomic problems. Ann Eugenics 7:179–188
17. Fix E, Hodges J (1951) Discriminatory analysis: nonparametric discrimination: consistency properties. Technical Report 4, USAF School of Aviation Medicine
18. Fletcher R (1987) Practical methods of optimization. Wiley, Chichester
19. Fogel I, Sagi D (1989) Gabor filters as texture discriminator. Cybernetics 61:103–113
20. Fukunaga K (1990) Introduction to statistical pattern recognition, 2nd edn. Academic, New York
21. Fukunaga K, Mantock J (1983) Nonparametric discriminant analysis. IEEE Trans Pattern Anal Mach Intell 5(6):671–678
22. Gentle J (1998) Numerical linear algebra for applications in statistics. Springer, Berlin
23. Haasdonk B (2005) Feature space interpretation of SVMs with indefinite kernels. IEEE Trans Pattern Anal Mach Intell 27(4):482–492
24. Haasdonk B, Keysers D (2002) Tangent distance kernels for support vector machines. In: Proceedings of the 16th ICPR, pp 864–868
25. Haasdonk B, Bahlmann C (2004) Learning with distance substitution kernels. In: 26th Pattern Recognition Symposium of the German Association for Pattern Recognition (DAGM 2004). Springer, Tübingen, Germany
26. Ju Han, Kai-Kuang Ma (2007) Rotation-invariant and scale-invariant gabor features for texture image retrieval. Image Vis Comput 25(9):1474–1481
27. Trevor Hastie, Robert Tibshirani (1996) Discriminant adaptive nearest neighbor classification. IEEE Trans Pattern Anal Mach Intell 18(6):607–616
28. Heiser W (1995) Convergent computation by iterative majorization: theory and applications in multidimensional data analysis. Recent advances in descriptive multivariate analysis, pp. 157–189
29. Huber P (1964) Robust estimation of a location parameter. Ann Math Stat 35:73–101
30. Kiers HAL (1990) Majorization as a tool for optimizing a class of matrix functions. Psychometrika 55:417–428
31. Krogh A, Hertz JA (1992) A simple weight decay can improve generalization. In: Moody JE, Hanson SJ, Lippmann RP (eds) Advances in neural information processing systems, Vol 4. Morgan Kaufmann, San Francisco, pp 950–957
32. Lawrence S, Giles C (2000) Overfitting and neural networks: conjugate gradient and backpropagation. In: Proceedings of the IEEE international conference on neural networks. IEEE Press, pp 114–119
33. De Leeuw J (1993) Fitting distances by least squares. Technical Report 130, Interdiviional Program in Statistics. UCLA, Los Angeles
34. Leibe B, Schiele B (2003) Analyzing appearance and contour based methods for object categorization. In: International conference on computer vision and pattern recognition (CVPR'03). Madison, WI, pp 409–415
35. Li Y, Shapiro LG (2004) Object recognition for content-based image retrieval. In: Lecture Notes in Computer Science. Springer, Heidelberg
36. Hsuan-Tien Lin, Chih-Jen Lin (2003) A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan. Available athttp://www.csie.ntu.edu.tw/~cjlin/papers/tanh.pdf
37. Mary X (2003) Sous-espaces hilbertiens, sous-dualités et applications. PhD thesis, Institut national des sciences appliquees de rouen - Insa rouen, ASI-PSI
38. David Masip, Ludmila I Kuncheva, Jordi Vitrià (2005) An ensemble-based method for linear feature extraction for two-class problems. Pattern Anal Appl 8(3):227–237
39. Tom Mitchell (1997) Machine learning. McGraw-Hill, New York
40. Moré JJ, Sorensen DC (1983) Computing a trust region step. SIAM J Sci Stat Comput 4(3):553–572
41. Moreno PJ, Ho PP, Vasconcelos N (2004) A kullback-leibler divergence based kernel for svm classification in multimedia applications. In: Thrun S, Saul L, Schölkopf B (eds) Advances in neural information processing systems, Vol 16. MIT, Cambridge
42. Nesterov Y, Nemirovskii A (1994) Interior Point Polynomial Methods in Convex Programming: Theory and Applications. Society for Industrial and Applied Mathematics, Philadelphia
43. Ong CS, Smola AJ, Williamson RC (2002) Hyperkernels. In: Neural information processing systems, Vol 15. MIT, Cambridge
44. Ong CS, Mary X, Canu S, Smola AJ (2004) Learning with non-positive kernels. In: ICML '04: Proceedings of the twenty-first international conference on Machine learning. ACM
45. R. Paredes, E. Vidal (2000) A class-dependent weighted dissimilarity measure for nearest neighbor classification problems. Pattern Recogn Lett 21(12):1027–1036
46. Rojas M, Santos S, Sorensen D (2000) A new matrix-free algorithm for the large-scale trust-region subproblem. SIAM J Optim 11(3):611–646
47. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290:2323–2326
48. Schölkopf B (2001) The kernel trick for distances. In: Leen TK, Dietterich TG, Tresp V (eds) Advances in neural information processing systems, Vol 13. MIT, Cambridge, pp 301–307
49. Shen L, Bai L (2006) A review on gabor wavelets for face recognition. Pattern Anal Appl 9(2–3):273–292
50. Smith JR, Chang S-F (1996) Tools and techniques for color image retrieval. In: Storage and Retrieval for Image and Video Databases (SPIF), pp 426–437
51. Squire D. McG, Müller W, Müller H, Raki J (1999) Content-based query of image databases, inspirations from text retrieval: inverted files, frequency-based weights and relevance feedback. In: The 11th Scandinavian Conference on Image Analysis. Kangerlussuaq, Greenland, pp 143–149
52. Tenenbaum JB, de Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290:2319–2323
53. Theodoridis S, Koutroumbas K (1999) Pattern recognition. Academic, London
54. Torkkola K, Campbell W (2000) Mutual information in learning feature transformations. In: Proceedings 17th international conference on machine learning, pp 1015–1022
55. Trafalis TB, Malyscheff AM (2002) An analytic center machine. Mach Learn 46(1–3):203–223
56. van Deun K, Groenen PJF (2003) Majorization algorithms for inspecting circles, ellipses, squares, rectangles, and rhombi. Technical report, Econometric Institute Report EI 2003-35
57. Vapnik VN (1995) The nature of statistical learning theory. Springer, New York
58. Vapnik VN (1998) Statistical learning theory. Wiley, New-York
59. Watanabe H, Yamaguchi T, Katagiri S (1997) Discriminative metric design for robust pattern recognition. IEEE Trans Signal Process 45(11):2655–2661

60. Webb A (1995) Multidimensional scaling by iterative majorization using radial basis functions. Pattern Recogn 28(5):753–759
61. Weiss S, Kulikowski C (1991) Computer systems that learn. Morgan Kaufmann, San Francisco
62. Zhou X, Huang T (2001) Comparing discriminating transformations and SVM for learning during multimedia retrieval. In: Proceedings of the 9th ACM international conference on multimedia. Ottawa, Canada, pp 137–146
63. Zhou X, Huang T (2001) Small sample learning during multimedia retrieval using BiasMap. In: IEEE computer vision and pattern recognition (CVPR'01), Hawaii

## Author Biographies

**S. Kosinov** received his M. Sc. degree in Computing Science from the University of Alberta, Edmonton, Canada in 2002. He then joined the Computer Vision and Multimedia Laboratory at the University of Geneva, Geneva, Switzerland, where he obtained his Ph.D. degree in Computer Science in 2006. Since then, he has held a graduate internship at the Advanced Media Management group at Intel Corp., Santa Clara, USA, and presently works at Google Inc. His research focus is on computer vision, machine learning and natural language processing topics.

**T. Pun** received his Ph.D. degree in image processing in 1982, at the Swiss Federal Institute of Technology in Lausanne (EPFL). He joined the University of Geneva, Switzerland, in 1986, where he is currently a Full Professor at the Computer Science Department and Head of the Computer Vision and Multimedia Lab. Since 1979, he has been active in various domains of image processing, image analysis, and computervision. He has authored or coauthored over 200 journal and conference papers in these areas as well as seven patents, and led or participated to a number of national and European research projects. His current research interests, related to the design of multimedia information systems and multimodal interaction, focus on data hiding, image and video watermarking, image and video content-based information retrievalsystems, EEG signals analysis, and brain-computer interaction.