Identity verification through finger matching: a comparison of Support Vector Machines and Gaussian Basis Functions classifiers

R. Brunelli

ITC-irst, Povo (TN), ITALY

Abstract

The paper presents a people identity verification system based on the matching of top view finger snapshots, supplementing purely geometrical finger shape comparison with textural information. Low dimensional feature vectors are used to train binary classifiers based on small Gaussian Basis Functions networks which, in this task, are able to match Support Vector Machines performance while outperforming them in runtime efficiency, thereby exposing a different facet in the comparison which complements available literature reports.

Key words: Support Vector Machines, Regularization Networks, Gaussian Basis Functions, Classification, Biometrics.

1 Introduction

Vision based biometrics systems have been gathering attention due to naturalness of interaction, reliability of operation, and steadily increasing performance/cost ratio. Theoretical and practical interest in the integration of multiple biometric traits for increased security has also been steadily increasing over recent years.

At the same time, the pattern recognition community has deepened her understanding of powerful classification paradigms such as Regularization Networks (RN) architectures (Poggio and Girosi, 1990; Evgeniou et al., 2000) and Support Vector Machines (SVM) (Vapnik, 1998; Schölkopf and Smola, 2001). This paper compares, on an experimental basis, Gaussian Basis Functions networks

Email address: brunelli@itc.it (R. Brunelli).

(Broomhead and Lowe, 1988), providing flexible data modeling and similar to RN architectures introduced by Poggio and Girosi (1990), with SVMs, whose classification performances have a solid theoretic foundation. The task chosen for the comparison is personal identity verification using dorsal images of fingers, coupling geometrical and textural information.

The reason for comparing these specific classifiers is twofold. On one side, RNs and SVMs have been shown to be related from a theoretical point of view, so that an experimental investigation of the relative merits of SVMs and of architectures similar to RNs, such as Gaussian Basis Functions networks, is of interest. On the other side, while several comparison have been made so far (B. Schölkopf et al., 1997; Moghadam and Yang, 2002), reporting an advantage for SVMs over standard Gaussian Radial Basis Functions networks, the conditions under which the comparison were performed do not allow to apply the same conclusions to more general Gaussian Basis Function (GBF) networks. This is particularly true for the choice of the expansion centers, their number and location, usually considered one of the strong points for SVMs. The experimental findings of the present paper are novel in that they show that Gaussian Basis Function networks, structurally similar to approximated RNs, may provide the same performance of SVMs with a very significant runtime computational advantage in complex applications.

The next section presents the data acquisition setup while Section 3 describes the feature extraction algorithms and the procedures for similarity assessment. After the introduction of a general classification framework in Section 4, the two techniques compared by the paper are presented. Finally, the performance demonstrated by the two algorithms is compared and commented upon in Section 5.

2 Data acquisition

Natural interaction is a key issue in the development of a biometric device. Many available hand recognition systems rely on constrained hand positioning to increase comparison accuracy and to reduce chances of unusable input data (Jain and Duta, 1999; Jain et al., 1999). The present study is based on relatively *unconstrained* hand positioning to increase system friendliness. While the resulting system provided some images that could not be appropriately processed (e.g. due to touching fingers) and thus required a limited number of additional interactions, system performance and interaction flow can be considered satisfactory, at least for some applications of interest. While the main purpose of this case study is the comparison of classification techniques, significant efforts have been put in the development of a system prototype that could be successfully deployed in the real world. Identity verification is

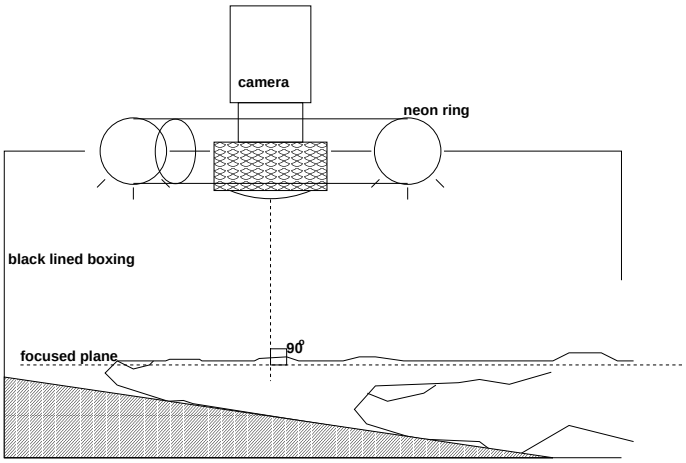


Fig. 1. The basic setup used for data acquisition. A cold light ring provides diffuse illumination without uncomfortable heating while camera is focused to make optimal use of available depth of field. The hand resting plane is slightly rotated so that the upper part of the hand is at (approximately) 90^o with camera optical axis, improving details readability. The interior of the box is completely covered with black velvet like material in order to reduce glare and to provide a black reference level. Camera electronics is set up so that black and white levels are automatically adjusted for.

based on the matching of *top view* finger snapshots, supplementing purely geometrical finger shape comparison with textural information.

The experimental setup is reported in Figure 1 and details are given in the corresponding caption. The system monitors the interiors of the box for significant changes. Whenever a sufficiently large area changes, the system waits for a static situation before triggering the acquisition of the image to be used for identity verification. Besides checking for a motionless image, the system verifies positioning within the camera field by looking at the center of gravity of the *hand* region. Additional controls ensure that the thumb and little fingers do not touch the box walls. Grey level images are acquired at a resolution of 578×706 pixels and 8 bits depth (see Figure 2A).

One hundred different persons interacted with the system. Each of them presented the right hand to the system several times, repositioning the fingers in different ways to provide varied inputs. A small subset of persons interacted with the system for several days to quantify time variability.

3 Feature extraction and comparison

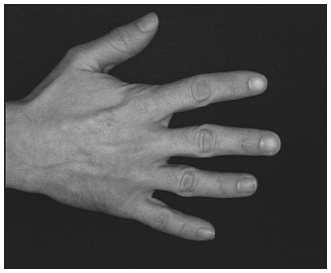
Once a promising hand image is grabbed, extraction of finger snapshots for comparison is done as follows:

(1) Detection of hand outline.
Relying on the geometry of the acquisition setup, the hand outline (see Figure 2B) is computed using the $A^\star$ algorithm (Nilsson, 1980). Outline boundary points are identified at the wrist sides on the open side of the box. The contour is followed using a merit function favoring the appropriate contrast at the boundary and penalizing abrupt direction changes.
(2) Finger extraction.
The hand outline determined by the edge following algorithm, is heavily low passed with a Gaussian filter. A plot of the corresponding distances from the middle of the wrist is reported in Figure 2C. The resulting path is approximated with a polyline computed with a predefined splitting threshold. The resulting number of segments must correspond to twice the number of fingers stored in the database for the given person. This enables the system to detect anomalous situations where some of the fingers are too close for proper segmentation, triggering interaction with the user for better hand positioning.
(3) Finger normalization.
Based on the information provided by the polygonal approximation step, each finger is cut at the palm junction, and its orientation estimated by the segment joining the middle of the cutting segment with the furthest point on the hand outline. The image is then rotated, clipped, and masked so that the resulting picture only contains information on the given finger (see Figure 2D).

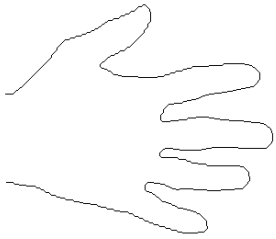
After successful detection of the fingers, a database entry is created for the given person, where all finger images are stored. Multiple image fingers can be stored for each user to improve matching performance. Runtime operation of the system requires the additional step of finger comparison to verify the identity claimed by the user. Many current hand recognition system rely solely on hand geometrical information completely discarding textural information (e.g. Jain and Duta, 1999). The system presented is instead based on both geometrical and textural information. Geometrical information is captured by the difference between finger profiles. Working on the normalized gray level image, the finger is explored longitudinally: the position of the left and right boundaries are located at the maxima of the gradient in the horizontal direction. The difference of two profiles is characterized by the area between them.

Textural similarity is assessed by the system through the use of the image correlation coefficient:

$$r = \frac{\sum_{xy}(A_{xy} - \bar{A})(B_{xy} - \bar{B})}{\sqrt{\sum_{xy}(A_{xy} - \bar{A})^2}\sqrt{\sum_{xy}(B_{xy} - \bar{B})^2}} \tag{1}$$

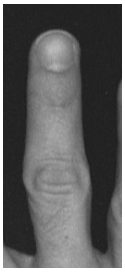


(A) Input image



(B) $A^\star$ contour following

(C) Contour distance maps



(D) Extracted finger

Fig. 2. The feature extraction work flow

where A_{xy} represents the gray value of pixel at coordinates (x, y) and $\bar{A}$ its average value over the compared template (similarly for B). Alternative, more robust, estimators proposed by Brunelli and Messelodi (1995) have been compared to r by means of the resulting receiver operating characteristic (ROC) curves, and found to be consistently less performing. In order to quantify the advantage, if any, of using textural over geometrical information, the ROC curve for a verification system using data from only the middle finger has been computed. The experimental data reported in Figure 3 support the usefulness of the *holistic* template matching approach and suggest that the extracted features do contain enough information for building a practical identity verification system.

Discarding thumb similarity (see Figure 4 and corresponding caption), a total of 8 values are available to build a vector $\boldsymbol{S}(h_i, h_j) \in \mathbf{R}^8$ quantifying the similarity of two hands h_i and h_j: 4 correlation scores $\{f_k^{ij}\}_{k=1,2,3,4}$ representing the textural similarity of each finger of hand h_i to the corresponding one of hand h_j, and the corresponding 4 profile distance values $\{p_k^{ij}\}_{k=1,2,3,4}$:

$$\boldsymbol{S}(h_i, h_j) = \left(f_1^{ij}, \ldots, f_4^{ij}, p_1^{ij}, \ldots, p_4^{ij}\right) \in \mathcal{F} \subset \mathbf{R}^8 \tag{2}$$

More sophisticated approaches could be used to automatically reduce the weight of unreliable features such as those presented by Brunelli and Falavigna (1995) and more recently, from a user-centric perspective, in Jain and Ross (2002), but their investigation lies outside the scope of the present paper.

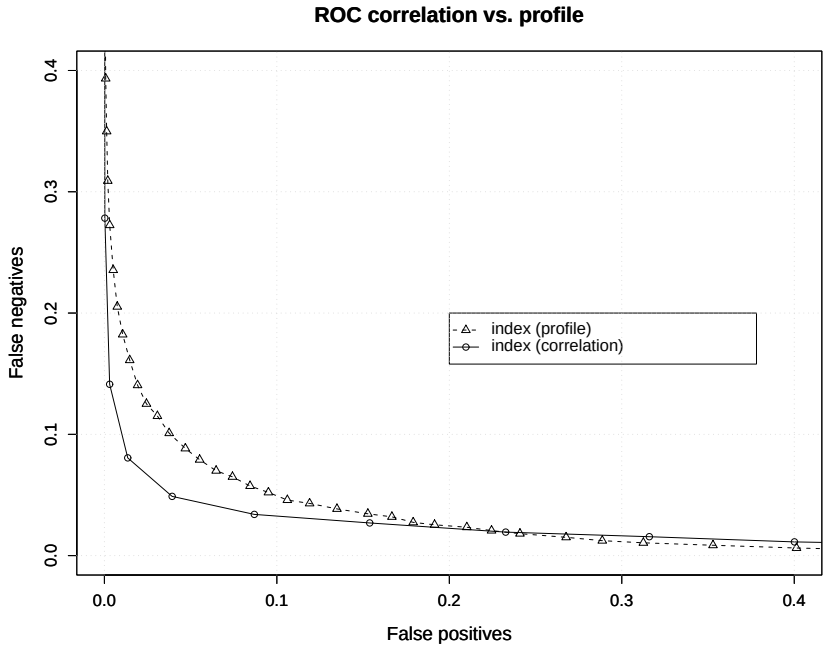


Fig. 3. Traditional hand comparison relies on a restricted set of parameters describing geometric features of the finger. In the proposed approach this information is augmented with finger textural information and the similarity is computed using the correlation coefficient. As the purpose of the plots is to show the relative discriminability of the two different features, the complete available dataset has been used (96 510 samples from 100 different people with a positive to negative ratio of 1 : 23).

The task to be solved is then to determine a map $F(\boldsymbol{x})$ such that:

$$F(\boldsymbol{S}(h_i, h_j)) = \begin{cases} -1 \text{ if } L(i) \neq L(j) \\ +1 \text{ otherwise} \end{cases} \tag{3}$$

where $L(i)$ represents the *identity label* of sample i. Let us note that map $F(\boldsymbol{x})$ itself has no index related to claimed identity: we are then not attempting to learn a user specific mapping (Jain and Ross, 2002).

Two different possibilities for increasing the available information at little computational cost have been investigated:

(1) Use of correlation values at multiple scales. The computation of image similarity using the correlation coefficient can be speeded up using a hierarchical approach based on a Gaussian image pyramid. At no additional cost, the entire set of values computed via hierarchical correlation, one similarity value per scale (see Figure 5), can be used instead of the single one at the highest resolution. The resulting similarity vectors exhibit detectable differences of scale behavior for the two classes to be discrim-

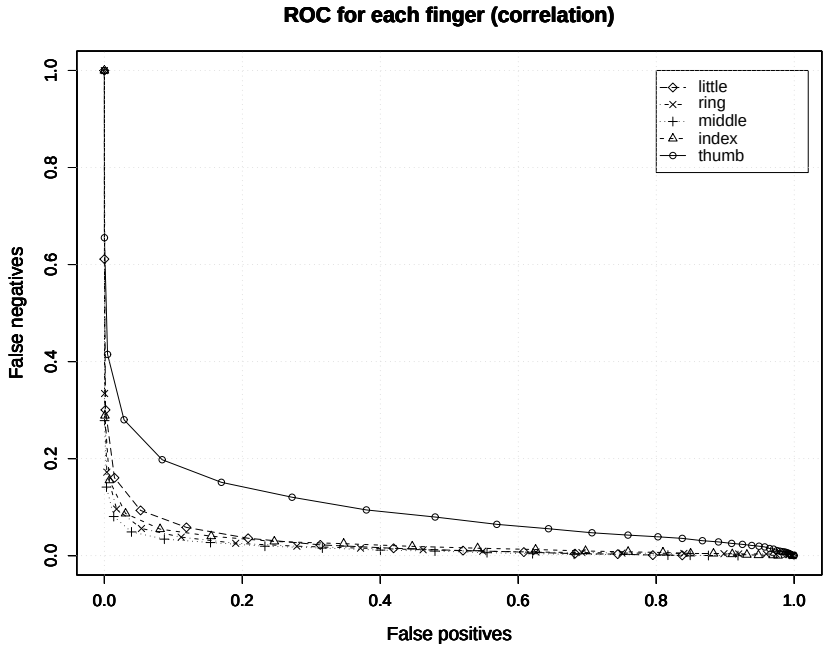


Fig. 4. Due to hand structural constraints, the value of the different fingers for identity verification varies. This is mainly due to the different poses assumed by thumb and little finger.

inated:

$$P(x_{r_1}|x_{r_2}, p) \neq P(x_{r_1}|x_{r_2}, n) \tag{4}$$

where p and n represent the positive and negative, i.e. impostor, class (see Figure 6).

(2) Extension of the set of scores by comparing the incoming images not only to those of the putative owner but also to those of a few other people known to be similar, constituting a *sentinel set* (see Algorithm 1). This approach has been first introduced for speaker verification in Higgins et al. (1991) under the name of *cohort model* and variations thereof adopted with success (among others Rosenberg et al. (1992); Mak et al. (2001); Barras and Gauvain (2003)). The resulting *sentinel scores* provide additional information characterizing the distribution of scores $\boldsymbol{S}(h_i, h)$ in the proximity of each given person i.

The distinguishing feature of our proposal is the usage of the complete set of cohort scores as input to the verification module, without summarizing them with a specific statistics (Rosenberg et al., 1992) or resorting to user specific modeling (Mak et al., 2001). Direct use of sentinel scores introduces significant person specific effects in the distribution of the values so that a normalization step, described in Algorithm 2, is necessary.

The performance of the two classification schemes considered in the paper is investigated using three feature spaces of different dimensionality d:

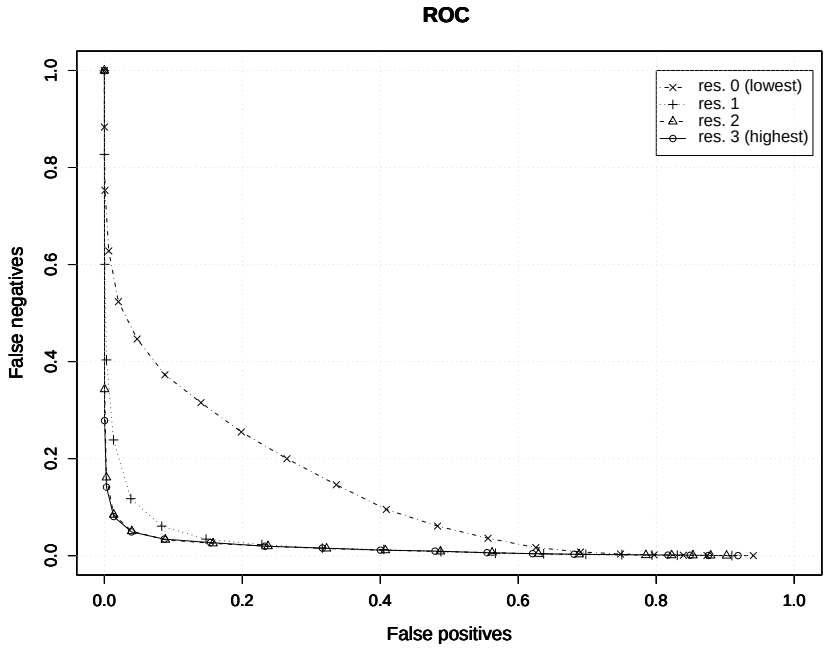


Fig. 5. Finger similarity is computed via hierarchical correlation to increase matching speed. Good results are obtained already at an average finger width of 16 pixels.

Algorithm 1: Computation of the sentinel set $\boldsymbol{G}(i)$

foreach *person i in database* **do**

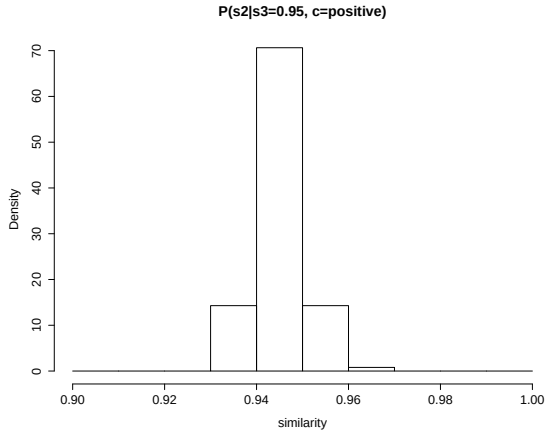
  foreach *other person j in database* **do**
    compute average similarity $\hat{S}_{ij}$;
  end
  sort $\left\{\hat{S}_{ij}\right\}$ by decreasing value;
  let $\alpha(\hat{S}_{ij})$ be the ranks ;
  $\boldsymbol{G}(i) = \left\{(j, \hat{S}_{ij}), j : 1 \leq \alpha(\hat{S}_{ij}) < 4\right\}$;
end

Algorithm 2: Usage of the sentinel set $\boldsymbol{G}(i)$

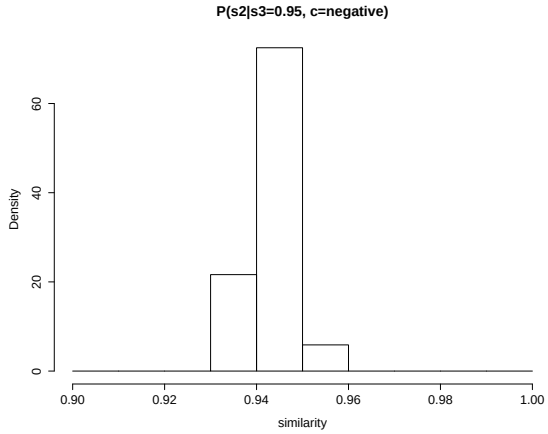
foreach *person j in $\boldsymbol{G}(i)$* **do**

  compute $S_{xj}/\hat{S}_{ij}$;
end
classify $\left(S_{xi}, S_{xj_1}/\hat{S}_{ij_1}, \ldots, S_{xj_3}/\hat{S}_{ij_3}\right)$

- single correlation score per finger, from the highest resolution image (`single`, $d = 8$);
- multi-scale score vector (`std`, $d = 20$);



positive



negative

Fig. 6. Given the correlation value s_3 measuring the similarity of two fingers at the highest resolution r_3, the distributions of the correlation values s_2 at a lower resolution differ for the two different classes: $P(s_2|s_3, p) \neq P(s_2|s_3, n)$. The plots show the two distributions for the case $s_3 = 0.95$, a high similarity value that would suggest a positive verification. Inspection of the two distributions shows that the expected value for s_2 is lower for negatives, a difference that may be exploited to improve classification accuracy.

- extended score vector based on the usage of a cohort score set (`sentinel`, $d = 32$ for three sentinels, using a single correlation score per finger);

The next section will consider the binary classification task corresponding to identity verification as a problem of non parametric estimation, addressing some critical training issues.

4 On classification

As formalized in Equation 3, the problem to be solved is that of approximating with a function f^* the unknown function F given its knowledge on a sparse data set $\{(\boldsymbol{x}_i, y_i)\}_{i=1,\ldots,N}$, represented by the similarity score vectors associated to hand images stored in the database. The resulting ill-posed problem may be turned into a well-posed one using regularization theory. Following Evgeniou et al. (2000), the approximation problem is transformed into a variational problem with the introduction of the functional $\mathbf{H}[f]$ which must be minimized by the approximating function f^*:

$$f^* : \mathbf{H}[f^*] = \min_{f \in \mathcal{H}} \mathbf{H}[f] \tag{5}$$

where

$$\mathbf{H}[f] = \frac{1}{N} \sum_{i=1}^{N} V_2(y_i, \boldsymbol{x}_i) + \lambda \|f\|_K^2 \tag{6}$$

$$V_2(y_i, \boldsymbol{x}_i) = (y_i - f(\boldsymbol{x}_i))^2 \tag{7}$$

and $\|f\|_K^2$ is a norm in a Reproducing Kernel Hilbert Space $\mathcal{H}$ defined by the positive definite function K, N is the number of data points and λ is the regularization parameter. It can be proved that, under quite general conditions, the solution of the variational problem can be expressed as

$$f^*(\boldsymbol{x}) = \sum_{i=1}^{N} c_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b \tag{8}$$

The error function in Equation 7 does not need to be quadratic, and can be tuned to the task considered. A classification problem would suggest:

$$V_+(y_i, f(\boldsymbol{x}_i)) = |1 - y_i f(\boldsymbol{x}_i)|_+ \tag{9}$$

where $|\cdot|_+$ is the so called *soft margin* loss function (Schölkopf and Smola, 2001) defined as:

$$|x|_+ = \begin{cases} x \text{ if } x > 0 \\ 0 \text{ otherwise} \end{cases} \tag{10}$$

An important result is that even in this case the solution has the form of Equation 8. Different classification techniques are associated to different choices of the error function V: Regularization Networks stem from V_2 while Support Vector Machines derive from V_+. The next sections will briefly contextualize these two approaches to the case study of the present paper, focusing on the training of the classifiers. A comparison of the two classifiers, addressing both

performance and run time efficiency is presented in the final section of the paper.

4.1 Classification through SVMs

Let us assume that we have a two class classification problem that is linearly separable. Among the separating hyperplanes, there is a unique optimal hyperplane characterized by having the maximum margin of separation between any training point and the hyperplane (Schölkopf and Smola, 2001). Working with hyperplanes corresponds to a particular choice of the kernel in Equation 8:

$$K(\boldsymbol{x}, \boldsymbol{x}') =< \boldsymbol{x}, \boldsymbol{x}' > \tag{11}$$

where $< \cdot, \cdot >$ denotes the dot product and formalization of the maximum margin requirement to the choice of V_+. An interesting result is that only a subset of the training points contributes to the expansion of Equation 8: the points lying at the minimal distance from the optimal hyperplane, also known as support vectors. The solution can be generalized to the case of non linearly separable problems using different RHKs: the value of $K(\boldsymbol{x}, \boldsymbol{x}')$ provides the scalar product of the input data mapped into a new feature space, related to the particular choice of the kernel. An example is given by the choice of polynomial kernels such as $< \boldsymbol{x}, \boldsymbol{x}' >^p$ which corresponds to mapping into a feature space whose dimensions are spanned by all possible p-th order monomials in the input coordinates. If the problem is non separable in the feature space, a *regularized* solution can still be obtained by allowing some points to be wrongly classified while *minimizing* the number of such points. The resulting minimization problem is given (Evgeniou et al., 2000) by:

$$H[f] = \frac{1}{N} \sum_{i=1}^{N} V_+ + \frac{1}{2C} \|f\|_K^2 \tag{12}$$

The solution is again of the form reported in Equation 8 and $0 \leq c_i \leq C/N$. Again, only a subset of the points will contribute to the expansion with a non null coefficient. In order to obtain a solution it is necessary

(1) to select an appropriate kernel, which identifies the actual classification feature space,
(2) to determine an appropriate *regularization* parameter C.

A popular choice in approximation theory is given by the Gaussian kernel:

$$K_G(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\gamma \|\boldsymbol{x} - \boldsymbol{x}'\|^2\right) \tag{13}$$

As the Gaussian kernel is a good choice also for the implementation of a classifier based on Gaussian Basis Functions, it was adopted for the classifica-

tion experiments. The classification problem has then two free parameters: the choice of γ in the definition of the kernel and the choice of C, the regularization parameter allowing for the non separability of the data.

In order to find the optimal values for the kernel parameters a hierarchical exploration of the parameter space has been conducted. The bi-dimensional parameter space has been sampled a first time using a coarse grid, which was then further refined in the proximity of the most promising value. For each considered point (γ, C) the performance of the corresponding SVM was evaluated in terms of total error (false positives and negatives over the total number of training samples) using 4-fold cross validation. Two alternatives to cross validation estimation of the optimal parameters for SVM kernels are analyzed in the paper by Chapelle et al. (2002). They are based on (approximations) to bounds and while the reported experimental evidence support their usefulness, they may not be generally useful. The performance of the SVM classifier as a function of kernel parameters and run time efficiency as expressed by the number of support vectors is addressed Figures 7-8.

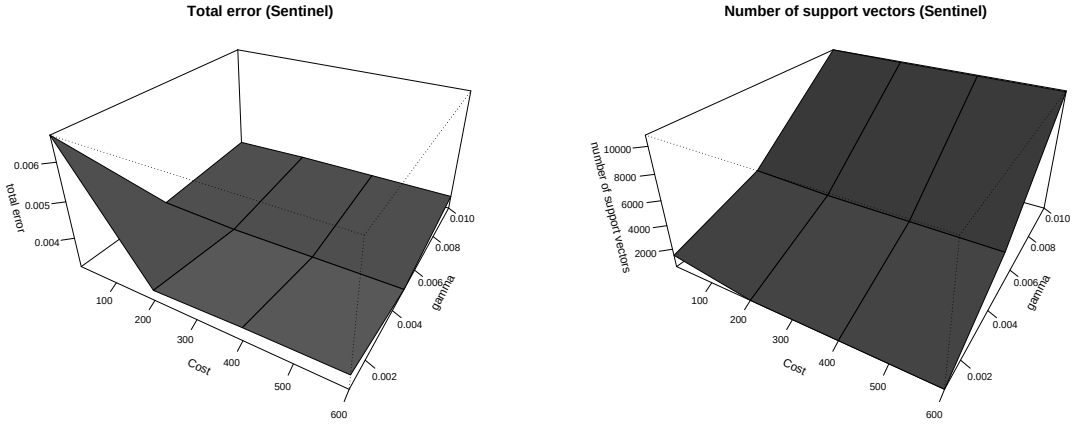


Fig. 7. The accuracy of the SVM classifier (left) depends on kernel parameters (i.e. γ) and slack control (i.e. C). In this particular case, due to the presence of a nearly flat region in the error surface, limited performance loss can be traded with reduced runtime computational complexity of the *trained* SVM that depends linearly on the number of support vectors found (reported on the right).

The plots of Figure 7 show that the number of support vectors can be a significant fraction of the number of available samples and that classification performance may depend significantly on the value of (γ, C): tuning of a SVM turns then out to be necessary and potentially time consuming. The feature space used, besides affecting overall classification accuracy, also impacts on the complexity of the final SVM as can be seen in Figure 8. Classification accuracy using the three feature spaces will be addressed in Section 5.

The experiments described were performed using `libsvm`, a freely available implementation of SVM for classification and regression applications (Chang

and Lin, 2001).

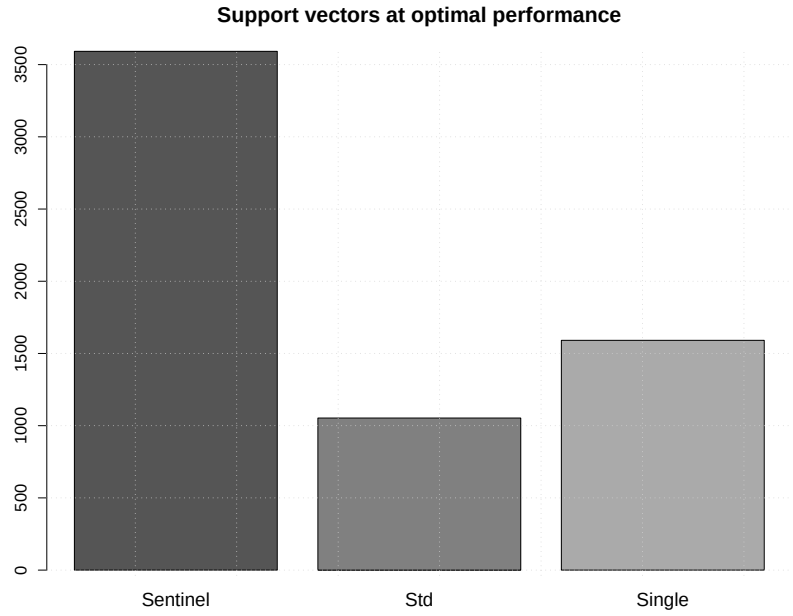


Fig. 8. The computational runtime complexity of the SVM depends on the number of support vectors at the point of optimal performance. As can be appreciated from the plot, the different input spaces result in very different operating complexities.

4.2 Classification through Gaussian Basis Functions Networks

Support Vector Machines have been originally introduced for classification and only later extended to regression. In this section we will leverage on Gaussian Basis Functions networks, structurally related to Regularization Networks, a regression scheme based on Equation 7 of which Radial Basis Function architectures are a well known example (Poggio and Girosi, 1990; Evgeniou et al., 2000; Cucker and Smale, 2002). The structure of Equation 8 can be interpreted in terms of a network structure with a single layer of hidden units. Different loss functions in Equation 7 result in different classification schemes and V_2 corresponds to regularized least square networks (Rifkin et al., 2003) of which Radial Basis Functions networks are an example. Let us note that in the V_2 case no sparsity can be expected and, in general, all units may participate in the reconstruction of the function, a fact that may pose computational difficulties. The accuracy of SVM and RN using Gaussian kernels has been recently compared on a suite of machine learning tasks (Zhang and Peng, 2004) and found to be essentially the same.

Mirroring the choice done in the SVM case, Gaussian kernels are chosen for the network. The Gaussian kernel has the property of being localized: it tends to

zero when $|\boldsymbol{x}-\boldsymbol{y}| \to \infty$. This behavior may be desirable in binary classification tasks using 0/1 as target values as the response of the network for inputs that differ significantly from the training patterns tend to be classified in class 0. In order to exploit this feature, the domain of the function to be approximated (see Equation 3) has been changed from $[-1, 1]$ to $[0, 1]$. Due to the localized response of the units, a Gaussian RBF network can be considered as a generalized look up table, and the value of γ in the kernel definition tunes the generalization of each unit. An expansion of the unknown function using less basis functions than examples can be found as an approximation of the solution of the variational problem by allowing centers to move (Poggio and Girosi, 1990). A possible limitation of this approach is given by the fact that the shape of the basis functions is fixed, i.e. it does not adapt to the data. In the case of Gaussian units this amounts to the usage of a uniform scaling γ. A superposition of basis functions characterized by different scales corresponding to a superposition of stabilizers in Eq. 6 can be justified within regularization networks (HyperBF networks) but would significantly expand the number of units needed, while still being constrained to a predefined discrete set of scales.

These observations suggest to depart from strict RNs and consider Gaussian Basis Function networks (Broomhead and Lowe, 1988; Bishop, 1995) based on the optimization of all unit centers and of all coordinates scaling during training by means of a spatially variant metric:

$$f^*(\boldsymbol{x}) = \sum_{k=1}^{n} c_k \exp\left(-\|\boldsymbol{x} - \boldsymbol{t}_k\|^2_{W_k}\right) \tag{14}$$

where f^* is the approximating function to be computed, n the number of network units ($n < N$), $\{c_k\}_{k=1,..,n}$ scalar values representing the weight of each unit in the expansion, $\boldsymbol{x}$ the input vector upon which the network must act, $\boldsymbol{t}_k$ is the vector representing the center of the k-th GBF unit, and W_k is a diagonal *scaling* matrix such that $\|\boldsymbol{x} - \boldsymbol{t}_k\|^2_{W_k} = (\boldsymbol{x} - \boldsymbol{t}_k)^T \boldsymbol{W}_k^T \boldsymbol{W}_k (\boldsymbol{x} - \boldsymbol{t}_k)$. The variables of the minimization problem of Equation 5, which we will solve under the assumption that $\lambda = 0$, leaving the framework of regularization, are then $\{c_k, \boldsymbol{W}_k, \boldsymbol{t}_k\}_{k=1,...,n}$. Let us note that this departs from commonly cited comparison of RNs and SVMs performance (e.g. B. Schölkopf et al., 1997; Moghadam and Yang, 2002) where centers are chosen using k-means clustering, a procedure blind to the specificity of the classification task, or derived from a support vector machine solution. The ill-conditioned aspect of the problem is now cured by the cross-validation choice of the number of network units. By using as many units as samples, minimization without regolarization may generate a network with zero error on the training set (a perfect look-up table of the training samples), and high error on the testing one, while by using a single unit, a high error on both training and testing sets is expected. A *good* network structure is determined within these two extrema by cross validation.

The associated training problem is highly non linear and presents multiple local minima, a reason for which GBF network in this form have been rarely considered. However, the similarity of their structure to mathematicaly well founded RNs and the additional modeling flexibility provided by locally adaptive scaling suggest robust performance with a limited number of units. The minimization problem associated to the training of the GBF network of Eq. 14 has been solved by means of the stochastic minimization algorithm described in Brunelli (1994) and Brunelli and Tecchiolli (1995). The algorithm is based on adaptive random search and constrained search time, automatically restarting itself in a new region of the search space when the extrapolated value of the current search fails to improve on the best computed solution. No explicit information on the gradient of the function to be minimized is needed and GBF classifiers based on a variety of loss functions can be trained. In a binary classification task the loss function is essentially a heuristic for finding the solution with the minimum expected number of classification errors. In fact, using the classification error as loss function makes the optimization problem hard: the function to be minimized is discontinuous with flat regions so that no gradient information helps in the minimization process. Training of neural and regularization networks has commonly adopted a quadratic loss function, leading to least square minimization (the reader is referred to Evgeniou et al. (2000) for a detailed account of the relationship of L_2 RN and SVM classification). Besides the commonly used L_2 and L_1 norm, two additional error functions have then been considered in an attempt to adapt network training to classification:

(1) cross-entropy (Bishop, 1995), derived by considering the network output as an estimate of posterior class probability:
(2) soft 01 step, as a continuous approximation to the classification error.

In the current case study, all network outputs are constrained to lay within $[0, 1]$ by mapping the GBF output through a sigmoidal unit making norms such as L_1 stricter than the usual L_2. The stochastic minimization algorithm used to solve Eq. 5 while flexible, is very computationally intensive. In order to decrease training time, a strategy for reducing the number of samples used has been adopted (Figure 9) resulting in shorter training time without any performance loss. Classification performance was found to exhibit only a slight dependence on the loss function used, slightly favoring the L_1 norm which was then chosen for experiments choosing the number of network units (see Figure 10, suggesting 32 as the optimal value).

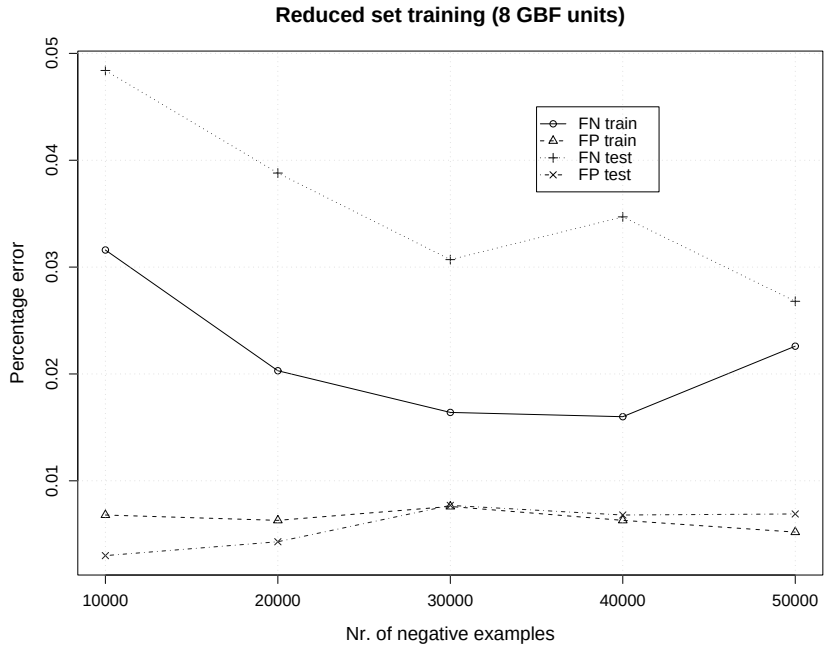


Fig. 9. A major issue regarding the usage of GBF (and neural) networks in classification tasks is that of training using large data sets. In order to ameliorate the situation the possibility of sub-setting the training set was tested. For each positive example, only a fraction of the corresponding negative examples was used. The latter were chosen as those whose average score was most similar to the positive examples set. Note the effect on the false positive error: being the train set significantly harder than the test set, the performance on the test set is actually better than that on the train set.

5 Results and Conclusions

The experiments reported in the previous section addressed the tuning of the two classifiers to the specific task: the current section focuses on the comparison of the accuracy and efficiency of two approaches. In order to determine algorithm performance, 4-fold cross validation has been employed using in turn 3/4 of the available data for training and the remaining part for testing. Four experiments have then been performed using randomly selected sets: care has been taken so that each cross validation experiment resulted in the same data sets for both (SVM and GBF) classifiers. Each of the 100 persons interacting with the system provided roughly 10 images, resulting $(10 \times 9)/2$ positive examples per person. Each of the four experiments in a cross validation session used 75 people for training and 25 for testing, resulting on average, in 2 981 (990) positive examples for training (testing) and 51 985 (40 554) negative examples.

The introduction of extended similarity score vectors (sentinel set) provided significantly better results for both classification schemes (see Figure 11). Us-

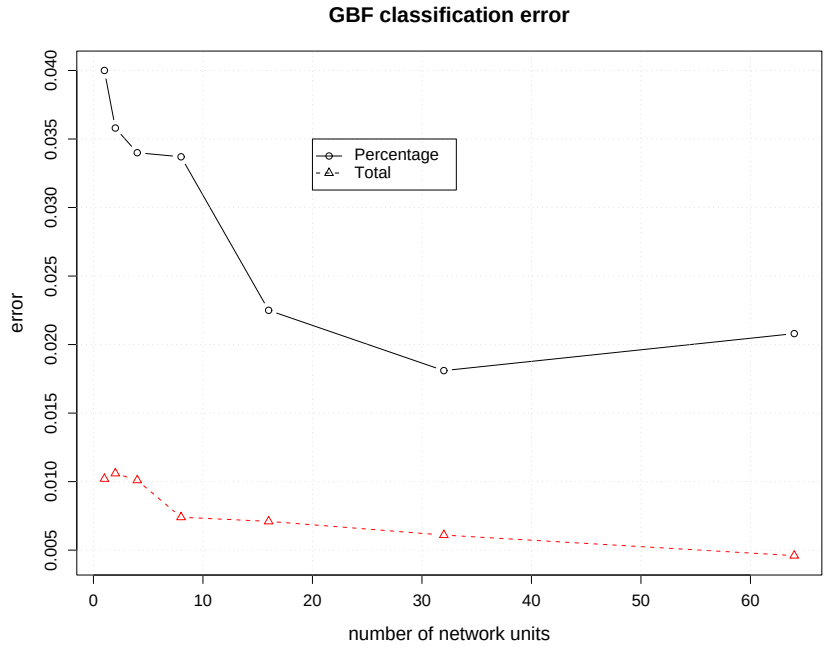


Fig. 10. The performance of the GBF network quickly saturates with the number of network units (no cohort information was used in this case).

ing correlation scores at different scales only benefited SVM: the GBF classifier using a single score shows a slightly higher accuracy than the SVM one using four scores, better exploiting available information through its more flexible modeling structure.

Table 1 presents the best performances obtained in the suite of experiments performed (SVM and GBF with sentinel set) in terms of false positive and false negatives.

Classifier	False pos.	Neg.	False neg.	Pos.	Tot. err.
SVM	95	40 554	36	990	0.00315
GBF	114	40 554	26	990	0.00337

Table 1
SVM with cohort was found to be the best performer by a very small margin. The table reports the average figures over a 4-fold cross validation experiment.

The performance attained with both classifiers suggests that finger template matching can be used for the development of a reliable identity verification biometric system, that may however need progressive update of stored images to avoid drifts of matching score distribution.

Let us now comment on run time efficiency of the classifiers considered. The full form of a SVM classifier using the Gaussian kernel of Equation 13 is given

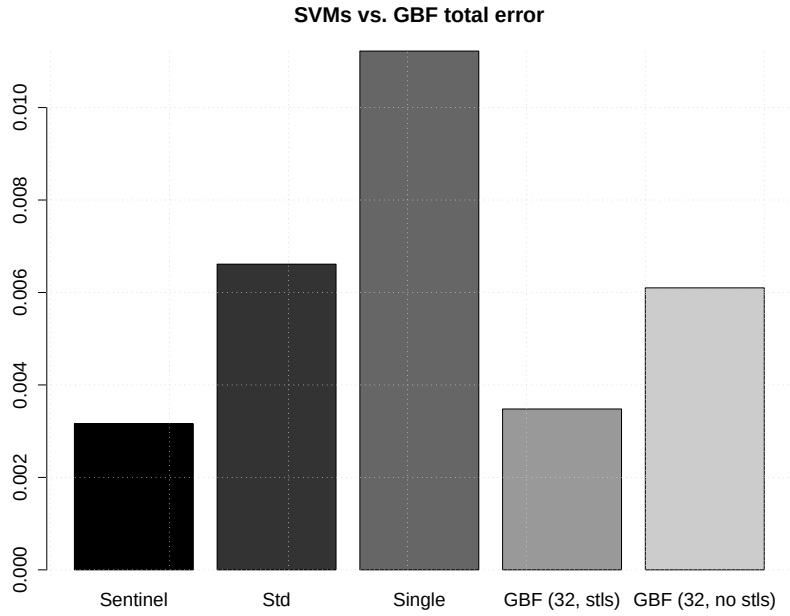


Fig. 11. The total error is given as the number of false positive and false negatives over the total number of testing examples. Adding sentinel information provides personalized information to the matching scores improving the performance of both classification schemes especially regarding false positive performance.

by:

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{n} c_i K_G(\boldsymbol{x}, \boldsymbol{x}_i) + b\right) \tag{15}$$

which, in the case of $\boldsymbol{x} \in R^d$ requires

$$o_{\mathrm{svm}} = 1 + 3n\,(d+1) \tag{16}$$

operations (where all operations, including exponentiation have been given the same weight). The number of operations o_{gbf} required for a GBF classifier (see Equation 14) is given by:

$$o_{\mathrm{gbf}} = 2 + 4n\,(d+1/2) \tag{17}$$

with the slight increase due to the usage of unit specific coordinate scaling. Table 2 reports the values of o_{svm} and o_{gbf} for two classes of experiments.

The experimental results support the following statement for the classification task considered:

> GBF network performance is comparable to the one that can be obtained with Support Vector Machines, providing a significantly better run time efficiency. The number of support vectors varies significantly, depending on kernel parameters and constraint control, even in regions where performance

Experiment	n_{svm}	o_{svm}	n_{gbf}	o_{gbf}	SVM/GBF
`sentinel`	3 592	355 609	32	4 162	85
`std`	1 053	28 432	32	1 090	26

Table 2
In the specific task considered, GBF classifiers outperform SVM classifiers in efficiency when working at (approximately) the same accuracy level.

exhibit small deviations from the optimal one, introducing the problem of efficiency tuning of SVMs (see Figures 7, 8).

The relative efficiency of GBF classifiers, due to the small number of network units required to match SVM performance, may be advantageous for the deployment of biometric systems relying on embedded devices, whose computational power is often limited, as well as for systems based on a client server architecture. The relatively unconstrained user interaction of the proposed system suggests its integration into user friendly multi-biometrics systems based on face and voice recognition.

Acknowledgments

The author would like to thank F. Bertamini at ITC-irst for his help in database acquisition and in the development of the software for finger boundary comparison.

References

B. Schölkopf, Sung, K.-K., Burges, C., Girosi, F., Niyogi, P., Poggio, T., Vapnik, V., 1997. Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classifiers. IEEE Trans. on Signal Processing 45 (11), 2758–2765.

Barras, C., Gauvain, J.-L., 2003. FEATURE AND SCORE NORMALIZATION FOR SPEAKER VERIFICATION OF CELLULAR DATA. In: Proceedings of ICASSP. Vol. 2. pp. 49–52.

Bishop, C. M., November 1995. Neural Networks for Pattern Recognition. Oxford University Press.

Broomhead, D. S., Lowe, D., 1988. Multivariable functional interpolation and adaptive networks. Complex Systems 2, 321–355.

Brunelli, R., 1994. Training Neural Nets Through Stochastic Minimization. Neural Networks 7 (9), 1405–1412.

Brunelli, R., Falavigna, D., 1995. Person Identification Using Multiple Cues. IEEE Trans. Pattern Anal. Mach. Intell. 17 (10), 955–966.

Brunelli, R., Messelodi, S., 1995. Robust Estimation of Correlation with Applications to Computer Vision. Pattern Recognition 28, 833–841.

Brunelli, R., Tecchiolli, G., 1995. Stochastic minimization with adaptive memory. Journal of Computational and Applied Mathematics 57, 329–343.

Chang, C.-C., Lin, C.-J., 2001. Training ν-Support Vector Classifiers: Theory and Algorithms. Neural Computation 13 (9), 2119–2147.

Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S., 2002. Choosing Multiple Parameters for Support Vector Machines. Machine Learning 46 (1), 131–159.

Cucker, F., Smale, S., 2002. Best choices for regularization parameters in learning theory: on the bias-variance problem. Found. Comput. Math. 2 (4), 413–428.

Evgeniou, T., Pontil, M., Poggio, T., 2000. Regularization Networks and Support Vector Machines. Advances in Computational Mathematics 13 (1), 1–50.

Higgins, A., Bahler, L., Porter, J., 1991. Speaker Verification Using Randomized Phrase Prompting. Digital Signal Processing 1, 89–106.

Jain, A., Duta, N., 1999. Deformable matching of hand shapes for user verification. In: Proceedings of ICIP '99. Kobe, Japan.

Jain, A., Ross, A., Pankanti, S., March 1999. A Prototype Hand Geometry-based Verification System. In: Proc. of Audio- and Video-Based Personal Identification. Washington D.C., USA, pp. 166–171.

Jain, A. K., Ross, A., September 2002. Learning User-specific Parameters in a Multibiometric System. In: Proc. ICIP. Rochester, pp. 57–60.

Mak, M., Zhang, W., He, M., May 2001. A New Two-Stage Scoring Normalization Approach to Speaker Verification. In: Proc. Int. Sym. on Intelligent Multimedia, Video and Speech Processing. Hong Kong, pp. 107–110.

Moghadam, B., Yang, M.-H., July 2002. Learning Gender with Support Faces. IEEE Trans. on PAMI 24 (7), 707–711.

Nilsson, N., 1980. Principles of Artificial Intelligence. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

Poggio, T., Girosi, F., 1990. Regularization Algorithms for Learning That Are Equivalent to Multilayer Networks. Science 247, 978–982.

Rifkin, R., Yeo, G., Poggio, T., 2003. Advances in Learning Theory: Methods, Model and Applications. Vol. 190 of NATO Science Series III: Computer and Systems Sciences. IOS Press, Amsterdam, Ch. Regularized Least Squares Classification.

Rosenberg, A. E., Delong, J., Lee, C.-H., Juang, B.-H., Soong, F. K., October 1992. The Use of Cohort Normalized Scores for Speaker Recognition. In: Proc. ICSLP-92. Banff, pp. 599–602.

Schölkopf, B., Smola, A. J., 2001. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press.

Vapnik, V., 1998. Statistical Learning Theory. Wiley, New York.

Zhang, P., Peng, J., 2004. Svm vs regularized least squares classification. In: ICPR (1). pp. 176–179.