Classification in a Normalized Feature Space Using Support Vector Machines

Arnulf B. A. Graf, Alexander J. Smola, and Silvio Borer

Abstract—**This paper discusses classification using support vector machines in a normalized feature space. We consider both normalization in input space and in feature space. Exploiting the fact that in this setting all points lie on the surface of a unit hypersphere we replace the optimal separating hyperplane by one that is symmetric in its angles, leading to an improved estimator. Evaluation of these considerations is done in numerical experiments on two real-world datasets. The stability to noise of this offset correction is subsequently investigated as well as its optimality.**

Index Terms—**Classification error, dataset partitioning, feature space, input space, noise, normalization, optimal separating hyperplane (OSH), optimality, support vector machines (SVMs).**

I. Introduction

SUPPORT vector machines (SVMs) have drawn much attention due to their good performance and solid theoretical foundations [1], [2]. Quite often, the use of SVMs on real-world datasets such as images implies the need for a preprocessing stage [2], [3]. Previous studies [4] have shown that *normalization* is a preprocessing type which plays an important role in SV classification. Using geometric considerations it is possible to exploit the normalization further by adjusting the threshold commonly used in SVMs.

This paper is structured as follows: Section II deals at first with normalization of the vectors in the input space. A problem inherent to this type of normalization related to SV classification is outlined and a solution, namely the normalization of the feature space, is subsequently presented, the latter being applied to the vectors of the feature space through normalization of the kernel function. An adaptation of the SV algorithm taking the normalized kernel functions into account is then presented: the position of the optimal separating hyperplane (OSH) is modified. A presentation of experimental results follows: binary classification is studied in Section III, and multiclass problems are discussed in Section IV. Here, we apply the estimator to an image dataset where the different types of normalization are compared as well as the stability to noise and the optimality of the OSH offset correction. The statistical significance of the proposed correction is studied in Section V, and Section VI concludes the paper.

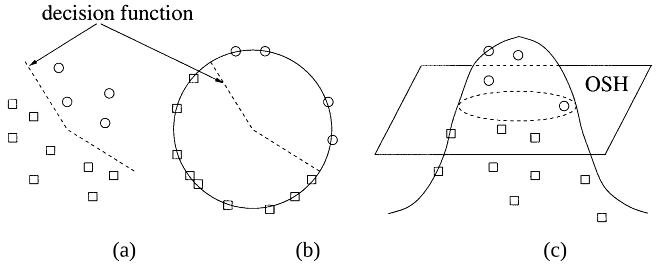

Fig. 1. (a) Two classes of vectors in a 2-D input space. (b) Normalization of these vectors such that they all lie on a unit hypersphere in input space. (c) Mapping of these vectors into the feature space.

II. SVMs in a Normalized Feature Space

Consider the most elementary type of preprocessing for SVMs, normalization of vectors $\vec{x} \in \mathbb{R}^N$ in *input space*. Here, the corresponding normalized vectors $\tilde{\vec{x}}$ are given by

$$\tilde{\vec{x}} = \frac{\vec{x}}{\|\vec{x}\|_2} = \left(\sum_{i=1}^{N} |x_i|^2\right)^{-1/2} \vec{x}. \tag{1}$$

The vector $\tilde{\vec{x}}$ lies on a unit hypersphere in $\mathbb{R}^N$. The SV algorithm is designed to find the OSH in the *feature space* which is obtained by a nonlinear mapping from the normalized input space. When considering the effect of such a mapping, in most cases we lose the normalization and/or scaling in the feature space as shown in Fig. 1. This may create a problem for the SV algorithm since it yields best classification performance for "input" vectors in the feature space which are in some way "scaled" [2], [3].

As suggested in [4], normalization in the feature space is a possible solution. Note that unlike in linear programming methods [5] where normalization acts on rows or columns of the design matrix only, normalization of the kernel functions can be considered as a simultaneous rescaling of rows and columns to obtain a matrix with all diagonal entries set to one (such methods are popular in numerical mathematics as matrix preconditioners). We, therefore, obtain the normalized kernel

$$\tilde{K}(\vec{x},\vec{y}) = \frac{K(\vec{x},\vec{y})}{\sqrt{K(\vec{x},\vec{x})K(\vec{y},\vec{y})}}. \tag{2}$$

Clearly $\tilde{K}(\vec{x},\vec{x}) = 1$, showing that all vectors in the feature space lie on a unit hypersphere. Furthermore, $\tilde{K}(\vec{x},\vec{y}) = \langle\tilde{\vec{\varphi}}(\vec{x}),\tilde{\vec{\varphi}}(\vec{y})\rangle$, where $\tilde{\vec{\varphi}}(\vec{x}) = \vec{\varphi}(\vec{x})/\|\vec{\varphi}(\vec{x})\| = \vec{\varphi}(\vec{x})/\sqrt{K(\vec{x},\vec{x})}$ stands for the "normalized" mapping. Clearly, normalized kernels satisfy Mercer's condition. In addition, the normalization of kernels is a conformal transformation [6] of the original kernels. Thus, the angles between vectors of the feature space are invariant with respect to normalization of the kernel functions [7]. In the case of single-class SVMs [8] with positive

Manuscript received September 17, 2001; revised August 30, 2002 and March 11, 2003. Part of this work was done while A. B. A. Graf was with the DI-LCN, Swiss Federal Institute of Technology, 1015 Lausanne EPFL, Switzerland in collaboration with S. Borer.

A. B. A. Graf is with the Max Planck Institute for Biological Cybernetics, 72076 Tübingen, Germany (e-mail: arnulf.graf@tuebingen.mpg.de).

A. J. Smola is with the RSISE, Australian National University, Canberra, ACT 0200, Australia.

S. Borer is with the Swiss Federal Institute of Technology, 1015 Lausanne EPFL, Switzerland.

Digital Object Identifier 10.1109/TNN.2003.811708

kernel functions, these normalized kernels play a predominant role. Indeed the angles between all vectors are then less than $\pi/2$ since $K(\vec{x},\vec{y}) > 0$ and, thus, the datapoints are placed on a portion of the same orthant on the unit hypersphere in the feature space. These vectors can then be more easily separated from the origin by a hyperplane.

For linear dot products, normalization in feature space and in input space is trivially equivalent, that is, (2) and (1) coincide. Gaussian RBF kernels $K(\vec{x},\vec{y}) = \exp(-\gamma||\vec{x}-\vec{y}||^2)$ are already normalized and, again, for monomial kernels $K(\vec{x},\vec{y}) = \langle\vec{x},\vec{y}\rangle^p$, normalization in input space is equivalent to normalization in feature space since

$$K(\tilde{\vec{x}},\tilde{\vec{y}}) = \langle\tilde{\vec{x}},\tilde{\vec{y}}\rangle^p = \left(\frac{\langle\vec{x},\vec{y}\rangle}{||\vec{x}||||\vec{y}||}\right)^p = \tilde{K}(\vec{x},\vec{y}). \tag{3}$$

It is well known that in the case of exponentiated dot product kernels [9] we obtain the RBF kernel as its normalization [10], i.e.,

$$\frac{\exp(\langle\vec{x},\vec{y}\rangle)}{\sqrt{\exp(\langle\vec{x},\vec{x}\rangle)\exp(\langle\vec{y},\vec{y}\rangle)}} = \exp\left(-\frac{1}{2}||\vec{x}-\vec{y}||^2\right). \tag{4}$$

Normalization in feature space does not only change the kernel functions but also affects the optimization problem (see also [11] for a first approach). Conventionally, the SV algorithm determines the OSH, given by its normal vector $\vec{w}$ and offset b, by a maximum margin construction. This means that the margins of separation are symmetric around the OSH since both lie at a distance $\delta = 1/||\vec{w}||$.

However, when considering a normalized feature space, all datapoints lie on a unit hypersphere. It would, thus, be more accurate to do classification not according to an OSH computed such that the margins are symmetric around it, but according to an OSH determined such that the margins define equal distances *on* the hypersphere. This is achieved by adjusting the value of the offset b of the OSH. The normal vector $\vec{w}$ remains unchanged after such a redefinition of the margin (see below).

In order to compute the correction to the value of b, consider Fig. 2. The intersection of the two margin hyperplanes with the unit hypersphere are parameterized by the angles α_1 and α_2 defined by $\cos(\alpha_1) = \delta(b-1)$ and $\cos(\alpha_2) = \delta(b+1)$. The bisection of the angle formed by α_1 and α_2 is represented by the angle φ and can be computed as follows:

$$\varphi = \frac{\alpha_1+\alpha_2}{2} = \frac{1}{2}\left(\arccos(\delta(b-1)) + \arccos(\delta(b+1))\right). \tag{5}$$

Moreover, the new position $\hat{b}$ of the OSH is defined by $\cos(\varphi) = \hat{b}$, yielding the following:

$$\hat{b}(\vec{w},b) = ||\vec{w}||\cos\frac{\arccos\left(\frac{b-1}{||\vec{w}||}\right) + \arccos\left(\frac{b+1}{||\vec{w}||}\right)}{2}. \tag{6}$$

The arguments of $\arccos$ in (6) are within range, as for the separation of data on the unit sphere $|b \pm 1|$ cannot exceed $||\vec{w}||$. The correction of b mentioned here leads to a new OSH defined by $(\vec{w},\hat{b})$. This method is valid regardless of the kernel function as long as it is normalized according to (2).

Fig. 3 shows that the correction to the offset of the OSH is important for large values of b. This is to be expected, since large values of b place the margins on the top (or bottom) of the hypersphere where it has a flat shape and is, thus, more sensitive.

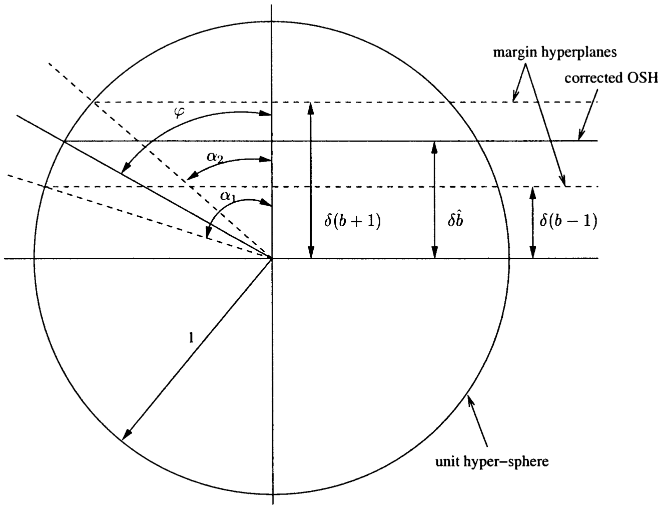

Fig. 2. Computation of the offset correction $\hat{b}$ for the OSH in a normalized feature space using the margin hyperplanes.

For fixed b, we see that the magnitude of the correction term increases with δ, that is, the larger this margin, the bigger the corrective effect.

For a formal derivation define the change in the offset of the OSH as

$$\epsilon(\vec{w},b) = b - \hat{b}(\vec{w},b) \neq 0. \tag{7}$$

The optimization problem corresponding to the corrected OSH can be stated as follows. We retain the optimization constraints, yet use an offset correction dependent on b and $\vec{w}$. The position of the OSH is given by

$$\langle\vec{w},\vec{x}\rangle + b(\vec{w},b) = 0. \tag{8}$$

Since we still require that $\langle\vec{w},\vec{x}_+\rangle - \langle\vec{w},\vec{x}_-\rangle = 2$, $\vec{x}_\pm$ being the closest examples belonging to two different classes, maximizing $|\vec{x}_+ - \vec{x}_-||$ is equivalent to

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}||\vec{w}||^2 + C\sum_{i=1}^{p}\xi_i \\ \text{subject to} \quad & t_i\left(\langle\vec{w},\vec{x}_i\rangle + b - \epsilon(\vec{w},b)\right) \geq 1 - \xi_i \\ & t_i \geq 0 \text{ for all } 1 \leq i \leq p \end{aligned} \tag{9}$$

where $\epsilon(\vec{w},b)$ determines the amount of skew given in the constraints. What we show now is that independently of the choice of $\epsilon(\vec{w},b)$ the optimization problem is identical to the one given by the standard soft margin SVM.

For this purpose we assume that there exists an optimal value $\hat{\epsilon}$, obtained after solving the optimization problem. In this case, the remaining optimization problem in $\vec{w}$ and b still must be optimal. Therefore, we can rewrite (9) as follows:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}||\vec{w}||^2 + C\sum_{i=1}^{p}\xi_i \\ \text{subject to} \quad & t_i\left(\langle\vec{w},\vec{x}_i\rangle + b - \hat{\epsilon}\right) \geq 1 - \xi_i \\ & t_i \geq 0 \text{ for all } 1 \leq i \leq p. \end{aligned} \tag{10}$$

One can easily check that by replacing b with $\hat{b} := b + \hat{\epsilon}$ we obtain the classical SVM soft margin problem and, thus, $\vec{w}$ and $\hat{b}$ can be found by solving the standard procedures. All that remains to be done after the optimization is to compute $\epsilon(\vec{w},b)$.

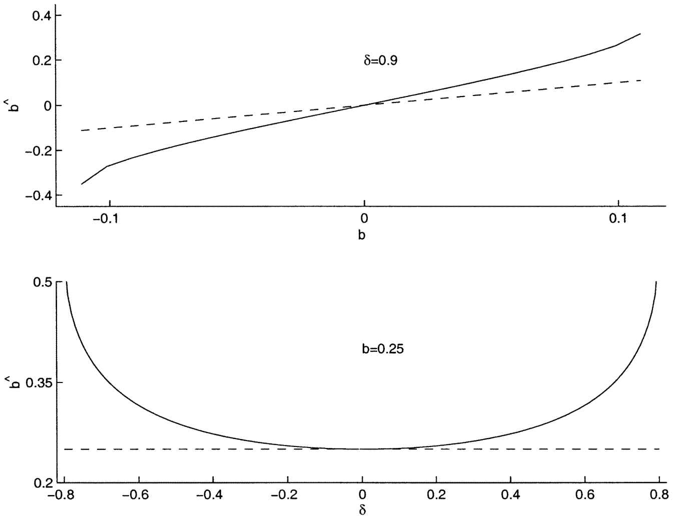

Fig. 3. Variation of $\hat{b}$ with respect to (top) b and (bottom) δ. The dashed line represents the uncorrected value b.

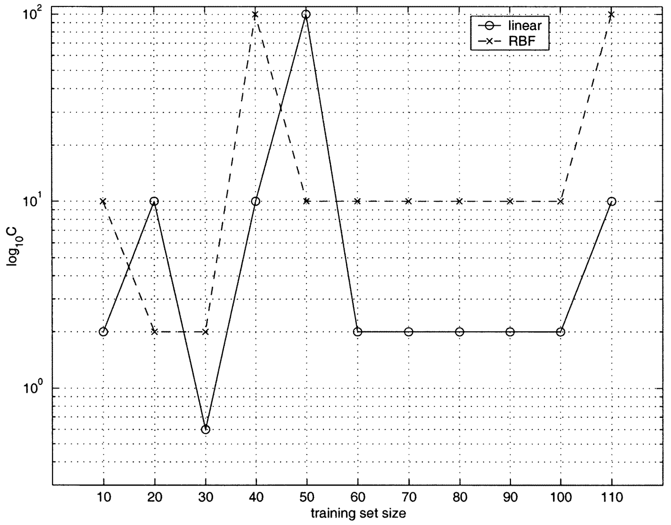

Fig. 4. Values of C as obtained by cross-validation on the training dataset as function of the size of the training set.

It would be interesting to develop the idea of distances on the hypersphere a bit further to include the penalty in the slack variables, rather than merely dealing with the margins. Unfortunately, this leads to rather unwieldy optimization problems, since then also ξ_i would have to be rescaled, thereby rendering the optimization problem nonquadratic. In online settings, however, such a modification could be easily accommodated for.

III. Binary Classification Experiments

To exemplify the properties we choose the ionosphere dataset from the UCI repository, available at http://www.ics.uci.edu [12]. We use the first 120 elements from this dataset and split the samples into training and test sets in increments of 10 while keeping the fraction of positive and negative examples constant. In other words, splits between training and testing partitions occur at: 10/110, 20/100, … 110/10. This way we can assess the sample dependency of the estimator. As optimizer we use SVM *light* [13] (version 3.50, which is available at www.kernel-machines.org). Finally, we consider linear kernels ($\langle \vec{x}, \vec{y} \rangle$) and RBF kernels ($\exp(-\gamma\|\vec{x} - \vec{y}\|^2)$ with $\gamma = 0.01$) in a normalized feature space. The use of these kernel functions implies that it is not necessary to study input space normalization since for linear kernels, input and feature space normalization are equivalent and RBF kernels are already normalized.

For each partition the optimal value of the regularization parameter C is determined by cross-validation experiments on the training set. For this, ten random samplings of 30% of the elements of the training set are used for training and the remaining 70% of the training set are used for testing. The SV algorithm is considered with and without OSH offset correction. The corresponding classification errors are averaged over the number of samplings and the minimum of the so-obtained error curve indicates the optimal value of C. The value of C yielding a minimum classification error across the two algorithms (with and without OSH correction) is retained and plotted in Fig. 4 for each partition.

Using this optimal value of C, 20 random classification experiments on each dataset partition are performed with

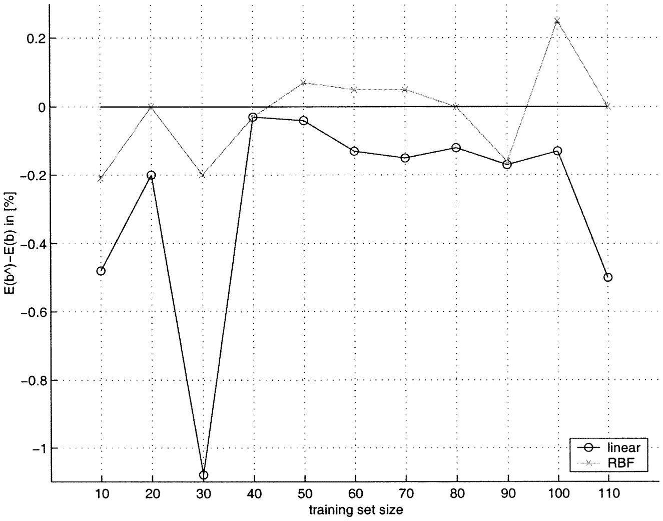

Fig. 5. Difference in classification error between the corrected and original OSH offset $E(\hat{b}) - E(b)$ as function of the size of the training set using the corresponding values of C.

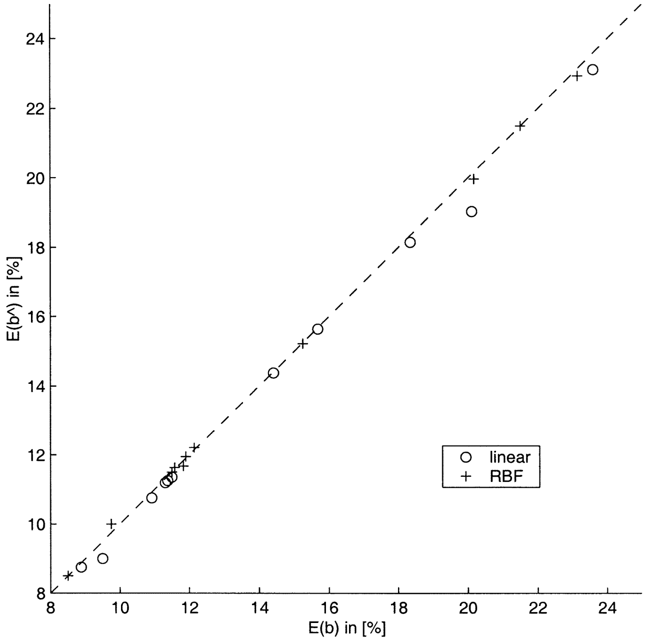

Fig. 6. Classification error with OSH correction as a function of the classification error without OSH correction, the dashed line bisecting the first orthant. The points represent the various partitions and the two kernel functions.

and without offset correction of the OSH, the corresponding differences of average classification errors being reported in Fig. 5.

The OSH offset correction is advantageous for all partitions when considering a linear kernel whereas for the RBF kernel, the correction is effective for small training sets. The mean relative correction of b measured as $(\hat{b}-b)/b$ yields on average over the dataset partitions 1.66% for the linear kernel and 0.49% for the RBF kernel. It can also be noticed that $\hat{b} < b < 0$ for both kernels and all partitions. The correction of the offset is thus less important for RBF kernels for the chosen dataset. In order to test the generalization behavior of the offset correction, the classification error with offset correction is plotted against the one without modification of the offset as shown in Fig. 6.

We may state that the offset correction performs slightly better than no correction for binary classification. Note that we are studying a symmetric classification problem, that is, the number of positive and negative examples is identical.

The multiclass experiments presented below do not exhibit this property. Moreover, the datasets are larger and the classification problem at hand is much harder. This provides a good ground for further studies of feature space normalization and OSH offset correction.

IV. Multiclass Experiments

For multiclass experiments we consider the COIL-100 dataset (Columbia Object Image Library, available at www.cs.columbia.edu). It consists of 128×128 pixels color images of

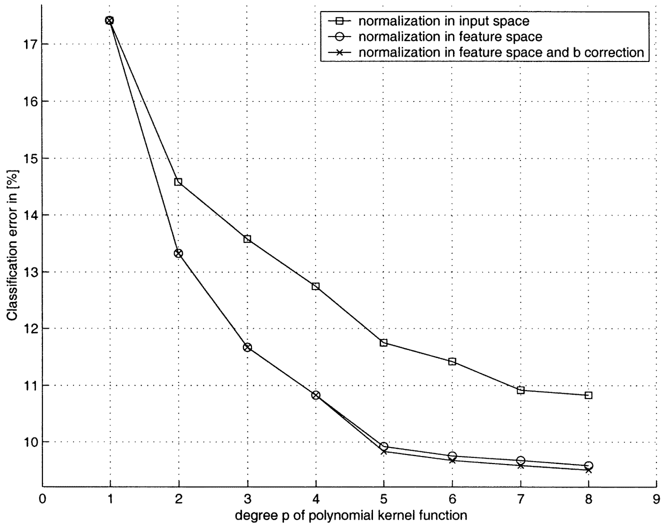

Fig. 7. Classification error with normalization in the input space, in the feature space and in the feature space with correction of b.

100 different objects, each one being represented by 72 views (one view for every 5° in a horizontal plane). The background of each image is black and the object is resized to the size of the image. To make the problem more challenging we downsample the images to 32×32 pixels greyscale images. The data are then separated into a training and a testing set, both being composed of regularly-spaced nonoverlapping perspectives of each object.

Cross-validation experiments over the various training sets (as given by different partitions and noise conditions, see below) show that the classification error is essentially insensitive to the choice of the tradeoff parameter C around a value of 1000, which is chosen below. Here, we consider polynomial kernels $((1 + \langle \vec{x}, \vec{y} \rangle)^p)$, since they tend to perform best on the data. Moreover, they are not *a priori* normalized, thus allowing the observation of the effect of a normalized feature space.

We use a simple one-against-rest method for multiclass classification (a collection of approaches is discussed in [14] for the same data). This means that we train 100 binary classifiers f_i and use the largest value of the real-valued estimators $f_i(\vec{x})$ to estimate the class label of a new pattern $\vec{x}$, i.e., $\hat{y}(\vec{x}) = \mathrm{argmax}_i f_i(\vec{x})$.

A. Effect of Normalization

We first check the influence of normalization and offset correction on the generalization performance. Recall that for polynomial kernels normalization amounts to rescaling the intensity of the pixels of the images (3).

The data set is partitioned as follows: the 12 perspectives 0-30-60-,…,-330 go into the training set and the 12 perspectives 15-45-75-,…,-345 into the testing set. The results presented in Fig. 7 show that normalization (except for linear kernels) improves the estimates significantly and that the offset correction leads to a small additional improvement. Note that the effects of normalization become more pronounced as the degree of the polynomial function increases since there, slight differences in the intensity can lead to very different scales of values in $K(\vec{x}, \vec{y})$.

While this is sufficient evidence to consider normalization in feature space, the issue of OSH offset correction needs further investigation. We study this effect in further detail in the following sections under different training/testing set partitions and for various noise conditions on the input images.

B. Effect of Dataset Partitions

We use the splits of the dataset shown in Table I to investigate details concerning the OSH offset correction. These partitions allow a uniform interpolation between the training and testing sets except for the 6/8 partition. The difference in classification performance between the corrected and original OSH offset is given in Fig. 8.

The OSH offset correction performs better or as good as the standard OSH offset in almost all cases. Although slight, the effect of the offset correction on the classification error is thus almost always present (see Section V for a statistical analysis). However, no general tendency among the various partitions or degrees of the polynomial kernel function may be observed. In the following, we retain the 12/12 partition to assess the stability to noise since it has a stable behavior with increasing degrees of the polynomial kernel function.

C. Stability to Noise

The stability of the OSH offset correction to noisy inputs (small perturbations of the input dataset) is investigated in this Section i.e., we study the effects of a noised dataset on the decision function through the value of the classification error. The following three types of noise which are added to the COIL-100 dataset as shown in Fig. 9 are taken into account:

- speckle or multiplicative noise created by a uniformly distributed random variable of mean zero and given variance;
- Gaussian white noise of mean zero and given variance;
- “salt and pepper” noise where “on and off” pixel values are added to the original image with a given density.

The classification error curves of Fig. 10 are monotonically decreasing and exhibit a smooth behavior. The offset correction of the OSH seems thus to be stable to noise on the input data.

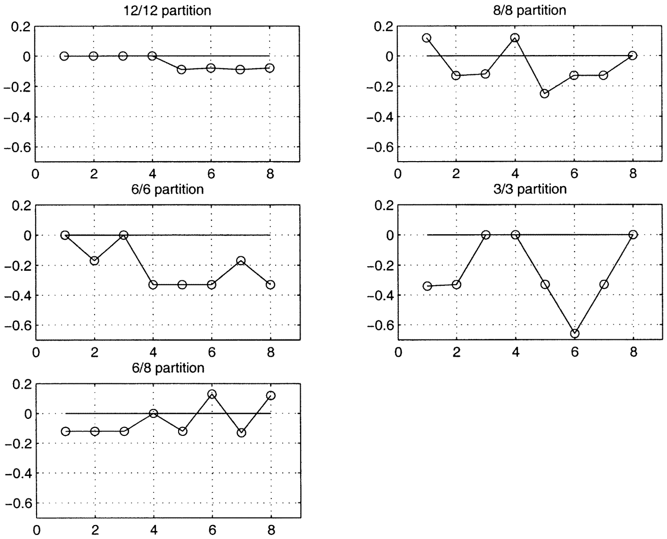

Fig. 8. Difference in classification error between the corrected and original OSH offset $E(\hat{b}) - E(b)$ in [%] as function of the degree of the kernel polynomial function for the considered partitions of the dataset.

TABLE I
Various Training and Testing Set Partitions and the Corresponding Views

train/test partition	training views	testing views
12/12 (as before)	0-30-...-330	15-45-...-345
8/8	0-45-...-315	15-60-...-330
6/6	0-60-...-300	30-90-...-330
3/3	0-120-240	60-180-300
6/8	0-60-...-300	10-55-...-325

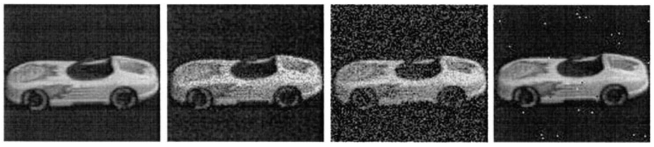

Fig. 9. From left to right: original image of the COIL-100 dataset, speckle noise (variance 0.04), Gaussian noise (variance 0.01 corresponding to $\sim 10\%$ noise) and salt and pepper noise (density 0.05 i.e., 51 pixels of the 1024 are affected).

As expected (see also [14]), the addition of noise increases the classification error. The added noise was, thus, too prominent to have helped during training [15], what may also be due to the built-in regularization of SVMs.

Furthermore, since the background of the original image is black, the same is true when speckle noise is applied to this image. Hence, it can be expected that the corresponding classification error is not much increased since only the object in the image is noised and not the whole image, the latter being classified by the SVM. However for the other types of noise, the background gets noised and a higher classification error can be expected. Finally, "salt and pepper" noise is most difficult to classify since the intensity of the pixel changes (either 0 or 255) are much stronger than with Gaussian noise, albeit not as frequent, which also results in using more SVs for classification.

To test the stability of the offset correction to noise, the classification error corresponding to an offset correction is compared to the one associated with no offset correction. Fig. 11 shows the improvement in generalization performance for various types and magnitudes of noise.

It may be stated that globally the offset correction of the OSH yields lower classification errors than the one corresponding to the uncorrected OSH. Moreover it can be observed that the offset correction gains importance as the magnitude of the noise increases, the latter being also accompanied by an increase of the classification error. For the "salt and pepper" noise, the error curves seem to have an oscillatory behavior regardless of the noise magnitude. This could be due to the fact that the "salt and pepper" noise affects strongly the classification performance of SVMs as seen in Fig. 10. Thus the oscillatory nature of these error curves may be due to the presence of too much noise. For the noise of highest density, the offset correction is most effective. For speckle noise, the classification error corresponding to the noise of intermediate variance can be considered as decreasing monotonically whereas the noise of low or high variance exhibit an oscillatory behavior with a peak in both cases for a polynomial function of degree 5. When considering Gaussian noise of low variance i.e., very localized noise, the offset correction seems not to ameliorate the classification performance of the SVM. However, for large variance and thus more blurring noise, the proposed correction clearly gets more advantageous as the variance of the noise increases. We notice, however, that in the above cases the error curves are not monotonically decreasing as the degree of the polynomial function rises.

D. Optimality of $\hat{b}$

Next we check the optimality of the proposed offset correction. For instance, for a normalized polynomial kernel of degree 8, we have $b \simeq 0.97\hat{b}$ as the average correction over the 100 classes. We then plot in Fig. 12 the classification error as function of various values of the offset b around its optimal value $\hat{b}$.

We see that this curve does not reach its minimum until $1.05\hat{b}$. Moreover, its behavior may roughly be fitted by a parabola with a global minimum at $1.07\hat{b}$. Since for all the classifiers we have

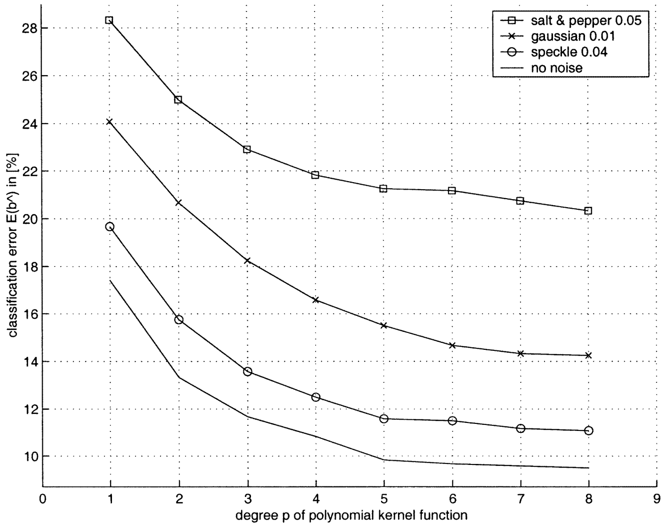

Fig. 10. Classification error for a normalized polynomial kernel function with OSH offset correction as function of the degree of the polynomial function without noise and with various types of noise on the input images: speckle (variance 0.04), Gaussian (variance 0.01) and "salt and pepper" (density 0.05) noise.

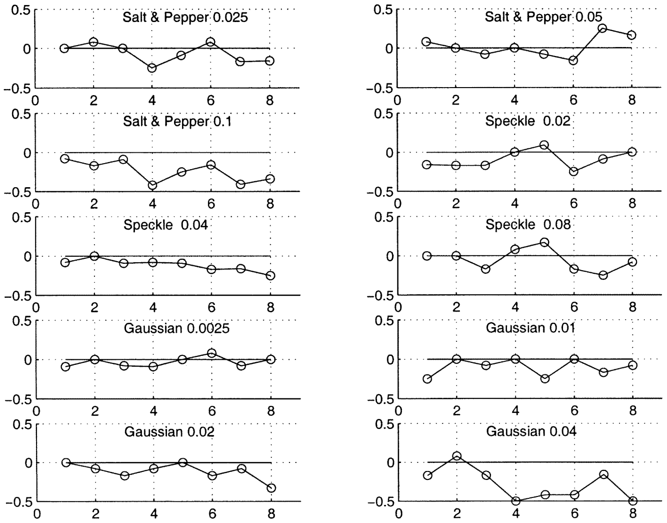

Fig. 11. Difference in classification error for a normalized polynomial kernel function between the corrected and the original OSH, $E(\hat{b}) - E(b)$ in [%], for various types of noise as function of the degree of the polynomial kernel function. The horizontal line represents the case of no improvement: $E(\hat{b}) = E(b)$.

$b < \hat{b}$, we suggest an overshoot of 5% and 7% from b toward $\hat{b}$ such that $b < \hat{b} < 1.05\hat{b} < 1.07\hat{b}$. When these overshoots are applied to the other degrees of the polynomial function, we get the plots of Fig. 13.

These error curves suggest that the overshoot of 5% is optimal for polynomial functions of degree 8 as could be expected since it was determined using the latter, but is not optimal for other degrees and even increases the classification error in some cases. No coherent behavior may be extrapolated from the two overshoots and neither one nor the other is optimal. Furthermore, we may compute the average difference between the error corresponding to an overshoot and the error corresponding to none over the degrees of the polynomial kernel function. We then get

$$\begin{cases} \frac{1}{8}\sum_{i=1}^{8}(E_i(1.05\hat{b}) - E_i(\hat{b})) = 0.041\% > 0 \\ \frac{1}{8}\sum_{i=1}^{8}(E_i(1.07\hat{b}) - E_i(\hat{b})) = 0.158\% > 0 \end{cases}.$$

Since all these means are positive, the correction of the offset of the OSH without overshoot is optimal *on average*. Overshooting can only increase optimality for a specific kernel function and has then to be determined empirically using error curves such as in Fig. 12. A parabolic interpolation of such error curves seems to be of no help when determining the best overshoot. When considering the linear kernel functions, we see that both overshoots exhibit much larger classification errors than without overshoot, corroborating the fact that globally overshooting is not appropriate.

V. Statistical Analysis of Offset Correction

We study here the significance of difference in classification error $E(\hat{b}) - E(b)$ for the ionosphere dataset (the 11 partitions and the two kernels) and the COIL-100 dataset (the five partitions and the ten types of noise for the eight kernels), yielding a total of 142 experiments. The differences themselves are of too

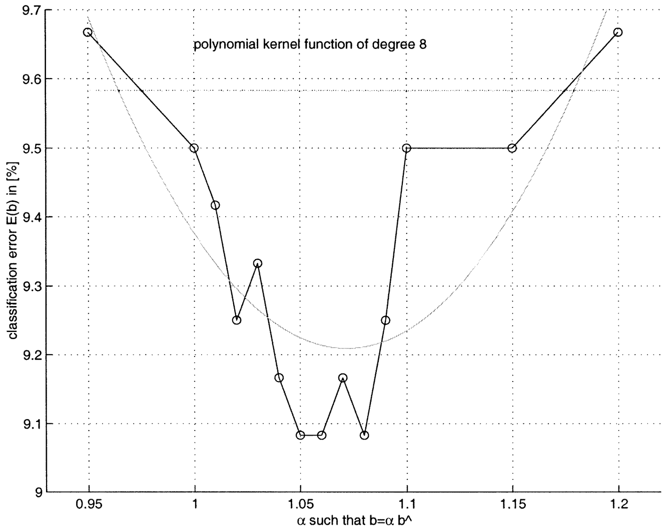

Fig. 12. Classification error for a normalized polynomial kernel function of degree 8 as function of the offset b. The latter is represented by the nondimensional coefficient α such that $b = \alpha\hat{b}$. The horizontal line represents the classification error with no offset correction. The curve represents a parabolic interpolation through the points and has a minimum at $\alpha \simeq 1.07$.

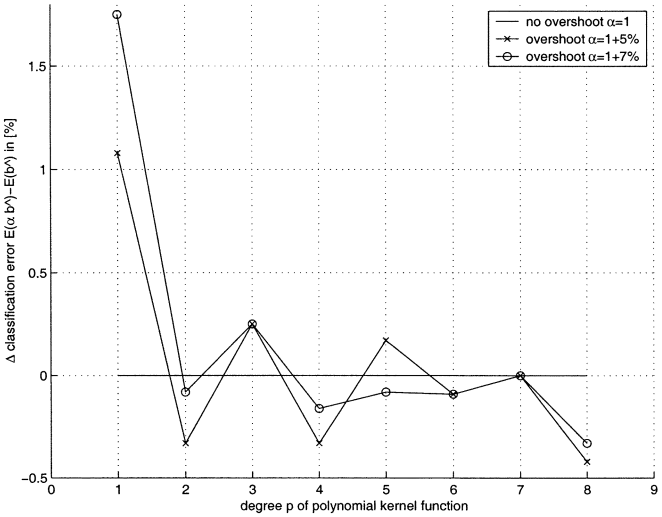

Fig. 13. Comparison between the classification error with OSH correction in a normalized feature space and the classification error obtained when considering an overshoot of 5% and 7% on the value of $\hat{b}$, $E(\alpha\hat{b}) - E(\hat{b})$, as function of the degree of the polynomial function of the kernel function.

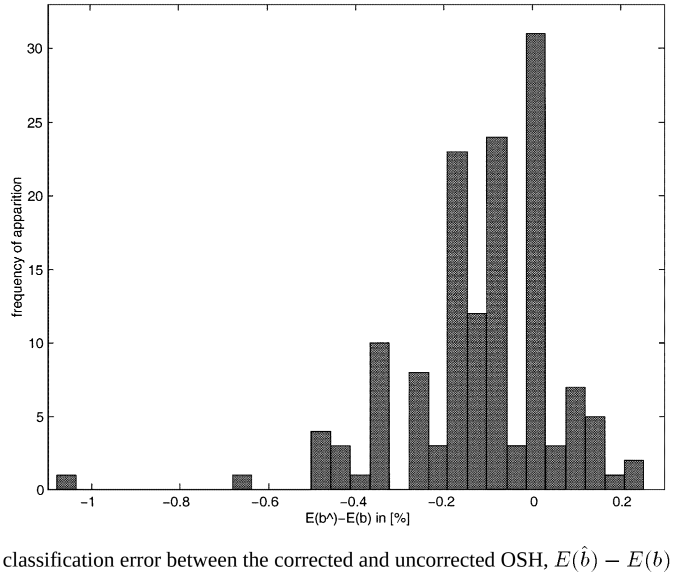

Fig. 14. Histogram of the difference in classification error between the corrected and uncorrected OSH, $E(\hat{b}) - E(b)$ over the 142 experiments.

small a magnitude to provide a significant result in a statistical test of dispersion (in, for example, a χ^2 test, or a signed dispersion test). However, based on the current results, a simple sign test is enough to show that the OSH correction method is advantageous in a significant number of cases. Of the total number of 142 comparative tests between $E(\hat{b})$ and $E(b)$ reported here, 31 yielded no difference. Of the remaining 111, 93 yielded a lower $E(\hat{b})$ than $E(b)$: this rate of 83.78% is highly significant given the null-hypothesis that $E(\hat{b}) > E(b)$ and $E(\hat{b}) < E(b)$ have equal probability ($P(E(\hat{b}) > E(b)) = 2.52e^{-9}$ from binomial distribution based on 111 trials with 0.5 probability of success). In the set of 142 comparisons, the observed range of differences $E(\hat{b}) - E(b)$ is from -1.08% to 0.25% as shown in Fig. 14. The histogram of this figure clearly indicates a higher frequency of occurrence of $E(\hat{b}) < E(b)$. Moreover, it can be noticed that for the cases where $E(\hat{b}) > E(b)$, the magnitude of this difference is about three times smaller on average than when considering the cases where $E(\hat{b}) < E(b)$.

VI. Conclusion

When dealing with SVMs, the normalization of the input data can influence dramatically the results of the classification, as well as the convergence of the SV algorithm. In this article the classification performance of SVMs in a normalized feature space is studied. Normalization in the input space was first discussed and it was noticed that it was not appropriate when considering SVMs since the feature space where classification is performed is then not normalized. A natural extension is thus to normalize the feature space, yielding normalized kernel functions. A careful analysis of the geometry of this normalized feature space suggests a modification of the position of the OSH. This novel algorithm has the same optimal solutions for $\vec{w}$ as the standard SV algorithm, but the considered correction is introduced in the final computation of the position of the offset b of the OSH. Numerical experiments corroborated that normalization in the feature space outperformed normalization in the input space and that the correction of the SV algorithm introduced in this paper was revealed to be most effective using a statistical analysis. It was shown experimentally that this correction is present for various datasets under various partitions and that it is stable to noise on the input data and is optimal on average. These considerations allow to conclude on a good generalization ability of the OSH offset correction.

Acknowledgment

The authors are grateful to C. Wallraven, J. Hill, and B. Schölkopf for helpful comments and suggestions. They also thank the anonymous reviewers who helped in strengthening the paper.

References

[1] C. Cortes and V. Vapnik, "Support Vector Networks," in *Machine Learning*. Boston, MA: Kluwer, 1995, pp. 273–297.

[2] V. Vapnik, *The Nature of Statistical Learning Theory*. Berlin, Germany: Springer-Verlag, 1995.

[3] S. Haykin, *Neural Networks: A Comprehensive Approach*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.

[4] R. Herbrich and T. Graepel, "A PAC-bayesian margin bound for linear classifiers: Why SVM's work," *Advances in Neural Information Processing Systems*, vol. 13, 2001.

[5] K. P. Bennett and O. L. Mangasarian, "Robust linear programming discrimination of two linearly inseparable sets," *Optimization Methods and Software*, vol. 1, pp. 23–34, 1992.

[6] S. Amari and S. Wu, "Improving support vector machine classifiers by modifying Kernel functions," *Neural Networks*, vol. 12, pp. 783–789, 1999.

[7] B. Schölkopf and A. J. Smola, *Learning with Kernels*. Cambridge, MA: MIT Press, 2002.

[8] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.

[9] B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds., *Advances in Kernel Methods*. Cambridge, MA: MIT Press, 1999.

[10] D. Haussler, "Convolutional Kernels on Discrete Structures," Comput. Sci. Dept., Univ. California, Santa Cruz, Rep. UCSC-CRL-99–10, 1999.

[11] A. Graf and S. Borer, "Normalization in support vector machines," in *Proc. DAGM 2001 Pattern Recognition*. Berlin, Germany: Springer-Verlag, 2001.

[12] C. L. Blake and C. J. Merz, "UCI Repository of Machine Learning Databases," Dept. Inform. Comput. Sci., Univ. California, Irvine, 1998.

[13] T. Joachims, "Making large-scale SVM learning practical," in *Advances in Kernel Methods—Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1999, ch. 11.

[14] M. Pontil and A. Verri, "Support vector machines for 3-D object recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 637–646, June 1998.

[15] C. M. Bishop, "Training with noise is equivalent to Tikhonov regularization," *Neural Comput.*, vol. 7, no. 1, pp. 108–116, 1994.

Arnulf B. A. Graf received the Dipl.Ing. degree in physics from the Swiss Federal Institute of Technology, Lausanne (EPFL), Lausanne, Switzerland, in 2000 and the DEA (master) degree in fluid dynamics at the Ecole Centrale de Lyon, Lyon, France, in 2000. He is currently pursuing the doctoral degree at the Max Planck Institute for Biological Cybernetics, Tübingen, Germany.

His current interests include the modeling of human classification and feature extraction using machine learning.

Alexander J. Smola studied physics at the University of Technology, Munich, Germany, at the Universita degli Studi di Pavia, Pavia, Italy, and at AT&T Research, Holmdel, NJ. He received the Master degree from the University of Technology in 1996 and the Doctoral Degree in computer science at the University of Technology, Berlin, Germany, in 1998.

He was a Researcher at the IDA Group of the GMD Institute for Software Engineering and Computer Architecture in Berlin. He is presently Leader of the Machine Learning Group, Research School for Information Sciences and Engineering, the Australian National University, Canberra, Australia. His research interests are kernel methods for prediction and data analysis. This includes classification and regression with support vector machines and unsupervised learning algorithms such as kernel principal component analysis, kernel feature analysis, and regularized principal manifolds.

Silvio Borer received the M.Sc. degree in mathematics from the University of Zurich, Zurich, Switzerland, in 1999. He is currently pursuing the Ph.D. degree in the Laboratory of Computational Neuroscience at the Swiss Federal Institute of Technology Lausanne, Lausanne, Switzerland.

His current research interests include pattern recognition and kernel algorithms.