

**SUPPORT VECTOR CLASSIFICATION OF IMAGES
WITH LOCAL FEATURES**

A Thesis Presented

by

MATTHEW B. BLASCHKO

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE

May 2005

Computer Science

© Copyright by Matthew B. Blaschko 2005

All Rights Reserved

SUPPORT VECTOR CLASSIFICATION OF IMAGES WITH LOCAL FEATURES

A Thesis Presented

by

MATTHEW B. BLASCHKO

Approved as to style and content by:

Erik G. Learned-Miller, Chair

Edward M. Riseman, Member

W. Bruce Croft, Department Chair
Computer Science

To Anne.

ACKNOWLEDGMENTS

Thanks to my family, friends, fellow grad students, and professors. A big thanks to Dima for suggesting kernels over local features. Thanks also to Jan Eichhorn for information regarding his implementation of the matching kernel.

This work was supported by the National Science Foundation under grant number ATM-0325167. For related work on classification of aquatic particles and additional information about the marine science data set used in this thesis, please visit <http://vis-www.cs.umass.edu/projects/bigelow/>.

ABSTRACT

SUPPORT VECTOR CLASSIFICATION OF IMAGES WITH LOCAL FEATURES

MAY 2005

MATTHEW B. BLASCHKO

B.S., COLUMBIA UNIVERSITY

M.S., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Erik G. Learned-Miller

The support vector framework is a general method for classification derived from inner products over feature vectors. The framework works by constructing maximal margin separating hyperplanes between classes. A key feature of this approach is that it allows for the replacement of strict inner products in the original feature space with *Mercer kernels*, functions that are equivalent to inner products between projections of the original vector into a higher, possibly infinite dimensional feature space. Though the data may not be well separated in the lower dimensional space, their projection into higher dimensions may be.

Kernels are especially interesting in their application to mathematical objects that do not lend themselves to be explicitly represented as a single vector in a finite dimensional space. Recent progress in the field of Computer Vision has relied on representations of images that consist of unordered sets of features that describe local image regions. I explore here several recently developed kernels between sets

of vectors and develop a common probabilistic theory to explain their design. This theory is based on a principled measure of similarity between the vector sets with few assumptions regarding structure in the data. Results are shown for several image data sets, including a challenging real-world marine science application. Results exceeding the current state of the art for bag of features representations are achieved with a newly proposed family of kernels. The best results to date on the VPR marine science data set are achieved with a kernel that combines local and global visual information.

TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS	v
ABSTRACT	vi
LIST OF TABLES	x
LIST OF FIGURES	xi
 CHAPTER	
1. INTRODUCTION	1
1.1 Local Features	2
1.1.1 SIFT Features	3
1.2 Support Vector Machines	5
2. APPROACHES TO KERNELS WITH LOCAL FEATURES	10
2.1 Multi-Instance Kernels	10
2.2 Kernel Principal Angles	11
2.3 Bhattacharyya Kernel	11
2.4 Matching Kernel	12
2.4.1 Choice of Minor Kernel	13
2.5 Pyramid Match Kernel	15
3. A PROBABILISTIC FRAMEWORK	17
3.1 Probability Product Kernels	17
3.2 Kernel Density Estimation	18
3.2.1 Probability Product Kernels Applied to Kernel Density Estimates	20

3.2.1.1	Example in Two-Dimensions	21
3.2.2	Bandwidth Selection	23
3.3	Similarities Between the Expected Likelihood Kernel with Density Estimation and Previous Approaches	24
3.3.1	Robustness Issues	24
4.	COMBINING MULTIPLE SOURCES OF INFORMATION	27
4.1	Multiple View Kernels	28
5.	EXPERIMENTAL RESULTS	31
5.1	Video Plankton Recorder Data Set	31
5.1.1	Comparison of Matching Kernel and Expected Likelihood Kernel	31
5.1.2	Robustness Experiments	35
5.1.3	Combining Local and Global Features	38
5.1.3.1	Sum of Local and Global Kernels	39
5.1.3.2	Product of Local and Global Kernels	41
5.1.3.3	Polynomial Combination of Local and Global Kernels	42
5.2	ETH-80 Data Set	43
6.	CONCLUSION	49
6.1	Future Work	50
	BIBLIOGRAPHY	52

LIST OF TABLES

Table		Page
5.1	Taxonomic categories of images in the VPR data set	32
5.2	VPR images used for distance experiments	35
5.3	Histograms of distances over all pairwise feature matches. Each row represents a different query image, while the columns are each a different target image.....	36
5.4	Histograms of distances over maximum feature matches. Each row represents a different query image, while the columns are each a different target image.....	37

LIST OF FIGURES

Figure	Page
1.1	Visualization of SIFT features in a marine science image. 4
1.2	The XOR problem is not linearly separable in the feature space, but is separable in the embedding space. 8
2.1	$\langle \{1, 1\}^T, \{x, y\}^T \rangle$ is plotted along with an iso-curve of $\ \{1, 1\}^T - \{x, y\}^T\ $, which result in a plane and a cylinder respectively. The averaging operation in equation (2.4) is difficult to interpret as we would expect to have equal contribution along the iso-curve. The equation is not coordinate free as changing the origin will result in different relative values for points selected by the argmin operation in equation (2.5). 14
3.1	Probability product applied to density estimations of two-dimensional data 22
5.1	A few example images from the VPR data set. 33
5.2	Bandwidth, σ , is plotted vs. accuracy on the VPR data set. Results are shown for the matching kernel, and for the expected likelihood kernel. 34
5.3	Fraction of minor kernel evaluations, β , is plotted vs. accuracy on the VPR data set. The kernel is computed as in equation 5.1. 38
5.4	Bandwidth, σ , is plotted vs. accuracy on the VPR data set. Results are shown for global features using a Gaussian RBF kernel. 39
5.5	Classification accuracies for the mean of the kernels for global and local features. 40
5.6	Classification accuracies for weighted averages of the kernels for global and local features. 41
5.7	Classification accuracies for the product of the kernels for global and local features. 42

5.8	Classification accuracies for an unweighted polynomial combination of the kernels for global and local features.	44
5.9	Classification accuracies for a weighted polynomial combination of the kernels for global and local features.	45
5.10	Example images from the ETH-80 data set.	46
5.11	Bandwidth, σ , is plotted vs. accuracy on the ETH-80 data set. Results are shown for the matching kernel, and for the expected likelihood kernel.	47
5.12	Fraction of minor kernel evaluations, β , is plotted vs. accuracy on the ETH-80 data set. The kernel is computed as in equation 5.1.	48

CHAPTER 1

INTRODUCTION

In object recognition, the two main components of feature extraction and classifier induction both have central importance. Vidal-Naquet and Ullman have shown that generic features with complex classifier induction and informative features with linear classification can both perform well on object recognition tasks [44]. Appropriate feature representations ease the burden on the classifier by providing greater class separation, and minimize structural risk, making it less likely that the classifier will overfit the data [43]. In some cases, features that do not lend themselves to representation in a single vector can add significant discriminability, but algorithms must be invented or adapted to account for the structure of the feature data.

Most object recognition systems tend to use either global image features, which describe an image as a whole, or local features, which represent image patches at interest points in the image. Global features have the ability to generalize an entire object with a single vector. Consequently, their use in standard classification techniques is straightforward. Local features, on the other hand, are computed at multiple points in the image and are consequently more robust to occlusion and clutter. However, they may require specialized classification algorithms to handle cases in which there are a variable number of feature vectors per image.

In designing classification algorithms for instances represented by multiple vectors, one could consider several approaches, each giving a different level of abstraction. Existing algorithms have often been based on bags of vectors, in which objects are represented by unordered sets of vectors. This is actually quite a restrictive ap-

proach as information regarding the location of the vectors in the image plane is completely ignored. Well designed classification systems that take into account this spatial information will likely perform better than one based solely on the bag of vectors approach. This work, however, focuses almost exclusively on approaches that do not take into account spatial information. This choice is based on the assertion that the techniques explored in this thesis can be the basis for systems that utilize additional information apart from this abstraction. By focusing only on the sets of vectors themselves, we develop a probabilistic framework for a principled comparison between two sets of vectors, which forms the basis of a comparison between two images. We then show how this framework can be extended to add additional sources of discriminative information.

Specifically, this work explores kernels between sets of vectors. A kernel is a function that relates two inputs, usually an indication of their similarity, and is central to several machine learning algorithms including Support Vector Machines (Section 1.2). By specifying a kernel between sets of vectors that is appropriate to the learning problem posed by local image features, we leverage the extensive existing literature on kernel-based learning (e.g. [39]).

1.1 Local Features

In practice, the generation of local features is commonly a two part process. The first requires an *interest point detector* to select points within the image that are located at visually distinctive patches. The second step generates a description of the region around that point. The data produced in these two steps consists of the location and description of the image patches around the interest points.

Interest point detectors look for visually distinctive regions in an image. The Harris corner detector is a simple example of such a detector, though a wide variety of choices exist in the literature [23, 33, 31, 30]. The most important aspect of an

interest point detector is repeatability, i.e. that the detector will select the same point on an object surface irrespective of minor changes in lighting and pose [35]. Interest point detectors typically detect features across different scales, and may incorporate affine invariance. In this case, the interest point detector provides information about the scale and normalizing transformation in addition to location within the image.

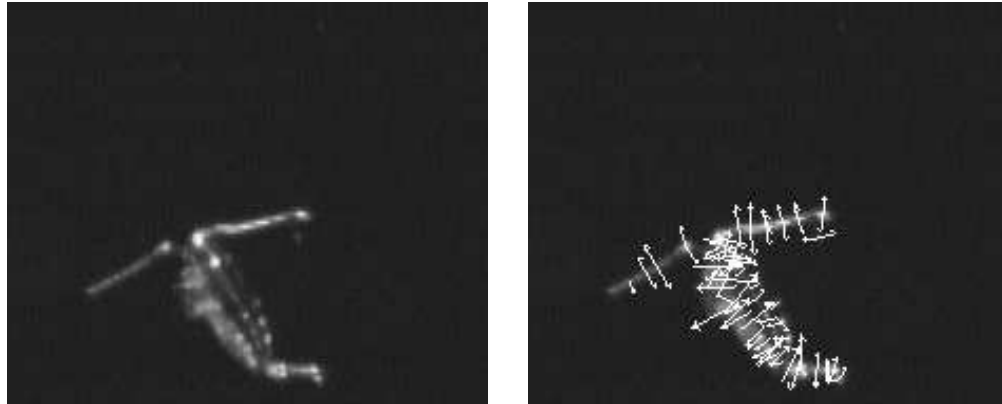
Local regions, most simply, could be described by a vector consisting of the intensity values around the image patch. Often, however, image patches are instead represented by statistics of the image gradient. Commonly proposed feature descriptors are histograms of the gradient orientation, or of the curvature of the intensity surface [37, 30]. Affine invariance can be achieved by selecting affine invariant statistics, or by transforming the image patch by the affine transformation estimated in the interest point detection phase [34]. Despite the variety of approaches in the interest point detection stage, classification algorithms based on local features tend to be influenced more by the descriptor than by the interest point detector [35].

1.1.1 SIFT Features

The Scale Invariant Feature Transform (SIFT) was originally proposed by Lowe in [29] and refined in [30]. The sift algorithm consists of four parts: (1) scale space extrema detection, (2) keypoint localization, (3) orientation assignment, and (4) keypoint description. The first step efficiently searches across scale space to find peaks in the gradient using a difference-of-Gaussian function

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) \star I(x, y) \quad (1.1)$$

where k is a constant multiple indicating the sampling rate in scale, \star is the convolution operation in x and y , and G is a two dimensional Gaussian. As Lowe points out, the difference-of-Gaussian is a close approximation to the Laplacian of Gaussian. Extrema are sampled at fixed scales for efficiency. The second step localizes



(a) Original image.

(b) SIFT features locations are shown by the tail of the arrows, scale is shown by the length, and dominant orientation by the direction.

Figure 1.1. Visualization of SIFT features in a marine science image.

keypoints to sub-pixel accuracy and throws out points that are not stable, leaving us with a candidate set of interest points. In the third step, dominant orientations of the keypoints are found based on the surrounding image patch. The image patches are rotated and scaled appropriately eliminating sensitivity to similarity transformations. The fourth step, identified by Mikolajczyk and Schmid to be the most important [35], is discussed in the following paragraph.

The final stage of the SIFT algorithm generates a description of the image patch based on the image gradient at the normalized local image patch. The goal is to create a representation that is simultaneously distinctive (i.e. image patches with different appearances will be placed far apart in feature space), while being robust to changes in illumination and camera position. Each patch is described by a three-dimensional histogram of gradient orientations, where two dimensions are the x and y location relative to the image patch, and the third dimension is the orientation itself. Lowe chooses 16 spatial bins, and 8 orientation bins for a total feature length of 128. The

histogram is normalized with a Gaussian weighting favoring the center of the 4 by 4 grid of spatial locations. A visualization of SIFT features on an example image is given in Figure 1.1. Improvements have been suggested that consist of using a log-polar spatial histogram rather than a rectangular histogram [35], or of using PCA to project the image patch down to a lower dimension while retaining distinctiveness of the image patch [24]. Nevertheless, SIFT features as described here are an extremely popular representation that has performed well in side by side comparisons of local feature descriptors [35]. Interestingly, clustering SIFT features tends to group visual primitives that correspond to the same object part [12].

1.2 Support Vector Machines

Support Vector Machines (SVMs) are classifiers that separate two class problems¹ with a maximum margin hyperplane [39]. In the case that the data are separable, the algorithm computes a hyperplane that separates the data while maximizing its distance to the nearest data point. A mechanical analogy is that two hyperplanes that are constrained to be parallel are placed between the exemplars of the two classes with a spring pushing them apart. These two hyperplanes cannot pass through any exemplar from either class. Once the spring has expanded as far as possible, the plane midway between the two planes and parallel to them is the decision boundary. This is also the plane that bisects the shortest line segment between the convex hulls of the two classes. In the case that the data are not separable, we introduce slack variables, ξ_i , to allow for some incorrectly classified exemplars.

More formally, the procedure for computing the maximizing hyperplane defined by

¹The extension to multi-class problems is explored in many works, many of which are independent of the Support Vector Framework. Some approaches include one vs. rest classification [38], pairwise classification [26], and error correcting output codes [10].

$$\langle w, x \rangle + b = 0 \tag{1.2}$$

where $w \in \mathbb{R}^D$, $b \in \mathbb{R}$, is given below, and $\langle \cdot, \cdot \rangle$ is the inner product. We define x_1, x_2, \dots, x_m to be the exemplars, and $y_i \in \{-1, 1\}$ the class labels. w is given in terms of its expansion

$$w = \sum_{i=1}^m \alpha_i y_i x_i \tag{1.3}$$

by the following quadratic programming problem:

$$\text{minimize}_{\xi, w, b} \quad \frac{1}{2} \langle w, w \rangle + C \frac{1}{m} \sum_{i=1}^m \xi_i \tag{1.4}$$

$$\text{subject to} \quad y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i, \quad i = 1, \dots, m \tag{1.5}$$

$$\text{and} \quad \xi_i \geq 0, \quad i = 1, \dots, m \tag{1.6}$$

We can in fact maximize the Lagrangian dual, in which the ξ_i disappear:

$$\text{maximize}_{\alpha \in \mathbb{R}^m} \quad \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \tag{1.7}$$

$$\text{subject to} \quad 0 \leq \alpha_i \leq \frac{C}{m} \text{ for all } i = 1, \dots, m, \tag{1.8}$$

$$\text{and} \quad \sum_{i=1}^m \alpha_i y_i = 0. \tag{1.9}$$

where each α_i is a Lagrangian multiplier. The decision function itself can then be written as a weighted sum of inner products where the weights correspond to the Lagrangian multipliers

$$f(x) = \text{sgn}\left(\sum_{i=1}^m y_i \alpha_i \langle x, x_i \rangle + b\right) \tag{1.10}$$

For additional details on the formulation of the quadratic programming problem, the reader is referred to [39]. Burges also provides an accessible tutorial introduction to Support Vector Machines [6].

A key feature of this framework is that the inner product, $\langle x_i, x_j \rangle$ in equation (1.7) can be replaced with a functional, $k(\cdot, \cdot)$, that is equivalent to a dot product in some space, $k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$. The choice $k(x_i, x_j) = \langle x_i, x_j \rangle$ is the simplest such assignment where we place a separating hyperplane in the original space, but in general, the pre-image in the original feature space of the decision boundary need not be linear. As a simple example, imagine that our data are in \mathbb{R}^2 . A mapping

$$\Phi(x) = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix} \quad (1.11)$$

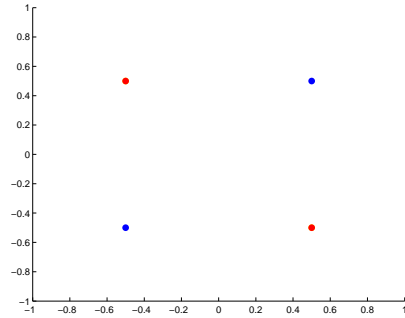
corresponds to a simple computation²

$$k(x_i, x_j) = (x_i \cdot x_j)^2 \quad (1.12)$$

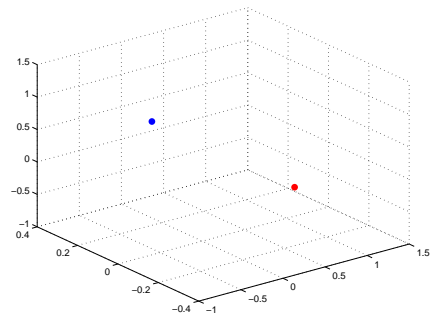
When used in a Support Vector Machine, the kernel will find a separating hyperplane in the transformed space, which consists of the power set of x . While there is no linear solution to the XOR problem (figure 1.2), the *second degree polynomial kernel* (equation (1.12)) is linearly separable in the transformed space. Furthermore, the simplified computation allows us to efficiently calculate this hyperplane without explicitly calculating the transformation, Φ . Similarly, we replace the inner product with a kernel evaluation in the decision function (equation (1.10)).

We can see that our choice of kernel allows us to add non-linearity to a decision function by computing a linear decision boundary in a transformed space. This space is often higher dimensional than the original input feature space. For example, in equation (1.11), vectors in \mathbb{R}^2 have been transformed to lie in \mathbb{R}^3 . Mercer's theorem tells us when a kernel is equivalent to an inner product in a transformed space

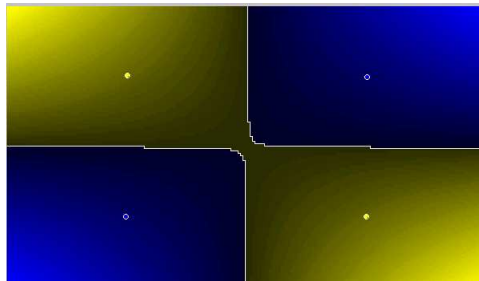
²In this case, the computation is a function of an inner product, but this is not necessary in general.



(a) The XOR problem is not linearly separable. Any separating hyperplane approach will fail on this data.



(b) The data are separable in the embedding space. Note that the two examples of each class both mapped to the same point due to symmetries in the data.



(c) The preimage of the second degree polynomial kernel is hyperbolic because noise was added to the coordinates prior to computation. In the limit, this will exactly partition the space into quadrants.

Figure 1.2. The XOR problem is not linearly separable in the feature space, but is separable in the embedding space.

without requiring us to explicitly formulate the transformation as we did in equation (1.11). Mercer's theorem states that to guarantee that $k(x_1, x_2)$ is equivalent to $\langle \Phi(x_1), \Phi(x_2) \rangle$ (i.e. that it is an inner product in some space), it is necessary and sufficient that the condition

$$\int_C \int_C k(x_1, x_2)g(x_1)g(x_2)dx_1dx_2 \geq 0 \quad (1.13)$$

be valid for all $g \in L_2(C)$, where C is a compact subset of \mathbb{R}^n [43]. Simply stated, any positive definite function is equivalent to a dot product in some space.

Several important classes of Mercer kernels have been developed for vector data. We have seen an example of a polynomial kernel in the preceding paragraphs. In general,

$$k_p(x_i, x_j) = (\langle x_i, x_j \rangle + 1)^p \quad (1.14)$$

is the p th degree polynomial kernel. The degree of the polynomial determines which powers of our original space will be represented in the higher dimensional space. Data that are not separable in low order terms of their polynomial expansions may be separated in higher order terms. We add 1 to the dot product in equation (1.14) in order to incorporate the lower order terms in our space as well. It is important to note, however, that it is quite easy to overfit as we can make very complex decision boundaries by setting the exponent too high.

Another important example of a family of Mercer kernels is the Gaussian radial basis function (RBF) kernel. A Gaussian RBF kernel returns to a constant factor the evaluation of a zero-mean Gaussian at a point given by the difference between two vectors.

$$k_\sigma(x_i, x_j) = e^{-\|x_i - x_j\|^2 / 2\sigma^2}. \quad (1.15)$$

The degree of smoothing, σ , determines at what scale we compare the data.

CHAPTER 2

APPROACHES TO KERNELS WITH LOCAL FEATURES

Recall that we are interested in characterizing images by possibly varying numbers of local features. To compare images, we would like to consider kernel functions that compare such sets of local features in two images. Recently, various techniques have been proposed for kernels between sets of vectors [15, 25, 46, 45, 17]. Of these, kernel principal angles [46], the Bhattacharyya kernel [25], and the matching kernel [45] have received some attention. Side by side comparisons between some of these approaches with respect to image recognition are available in [14] and [17]. Grauman and Darrell also formulate a fast algorithm for computing a kernel that bears some resemblance to existing kernels [17]. We describe each of the major techniques in the following sections, and evaluate some of their characteristics in the context of local image features.

2.1 Multi-Instance Kernels

Classification algorithms for multiple vectors include those generated for the multi-instance learning problem [11, 15], which was developed to predict the behavior of candidate chemicals for pharmaceuticals. This formulation, however, states that a set of vectors is considered to belong to a class if at least one member vector is a member of the class; only one vector need indicate membership while the rest can be viewed as noise, or irrelevant. Local features for image class recognition, on the other hand, can make use of many vectors in the set at once. Each vector ostensibly corresponds to a component of the object to be classified, and it is this collection of

components that indicates class membership. In classifying objects as belonging to a class, the presence of image features from many different points on the object surface are indicative of class membership, not the presence of only one of these features alone. By utilizing the set of vectors in its entirety, we can construct classifiers that are robust to partial occlusion and class variability. Consequently this kernel is not appropriate to the application of image classification as many visual cues should be employed rather than just one.

2.2 Kernel Principal Angles

Kernel principal angles was among the first kernels between sets of vectors to be proposed [46]. This technique computes a kernel based on principle angles between the subspaces spanned by the projections of the vectors in embedding space, $\Phi(x_i)$. Because the subspaces are constrained to be at most dimensionality equal to the number of vectors in the space, the authors restrict the kernel to vector sets of equal cardinality N . The kernel itself is defined to be

$$k(I, I') = \prod_{i=1}^N \cos(\theta_i)^2 \quad (2.1)$$

where θ_i is the i th principal angle between the subspaces. The primary weaknesses of the restriction to sets of equal cardinality, and the relatively poor performance in comparative studies [14] suggest that this kernel is of limited use to image classification.

2.3 Bhattacharyya Kernel

The Bhattacharyya kernel was proposed by Kondor and Jebara to parametrically represent the data in each set of vectors, and then use Bhattacharyya’s affinity (see section 3.1 for more details) between those parametric representations as the ker-

nel [25]. Specifically, kernel PCA [40] is used to fit a Gaussian to a set of vectors in an embedding space induced by the choice of an additional separate kernel, known as a minor kernel. The authors show how to compute Bhattacharyya’s affinity between Gaussians in closed form, which allows for exact computation of the kernel. However, this requires several matrix inversions, and the technique is among the most computationally demanding, being cubic in the number of vectors per set while most other techniques are quadratic, making it impractical for large sets of vectors despite its relatively high performance in comparative tests [14].

2.4 Matching Kernel

The Matching Kernel was proposed in [45] to handle sets of vectors resulting from interest point detectors and local image descriptors. A set of features results from the local descriptors which are then treated as sets of vectors describing the image. The kernel consists of a *minor kernel*, which is computed between individual vectors, and a function for combining the results of the minor kernel evaluations for the entire set. The function that computes the overall result takes the form

$$k(I, I') = \frac{1}{2}[\hat{k}(I, I') + \hat{k}(I', I)] \quad (2.2)$$

$$\hat{k}(I, I') = \frac{1}{N} \sum_{i=1}^N \max_{j=1, \dots, N'} \phi(x_i, x'_j) \quad (2.3)$$

where I and I' are sets of vectors corresponding to objects, x_i and x'_j are individual vectors in those sets, respectively, N is the number of vectors in I , and $\phi(x_i, x'_j)$ is the minor kernel. As [14] points out, equation (2.3) is not in fact positive definite due to the max operation, and so is not a Mercer kernel despite the claim in the original paper. Nevertheless, reported results in [45] and [14] indicate that this technique can be successfully applied to simple object recognition tasks.

2.4.1 Choice of Minor Kernel

Thus far we have not discussed what the choice of a minor kernel should be. There have been a variety of choices for minor kernels in the literature. Some of these do not seem to fit basic intuitive requirements for such a kernel, such as the fact the maximum kernel value for a point x_i be given with $x_j = x_i$. In the case of a dot product minor kernel, for a fixed query point $\mathbf{x} = \{x_1, \dots, x_n\}$, $\phi(\mathbf{x}, \mathbf{y})$, a function of \mathbf{y} , is a hyperplane that passes through the origin. Intuitively, we would like $\max_{j=1, \dots, N'} \phi(x_i, x'_j)$ to select a value for j that corresponds to a vector close to x_i in that space. Use of a simple dot product, however, will favor points that are infinitely far from x_i .

If we instead modify equation (2.3) to be

$$\hat{k}(I, I') = \frac{1}{N} \sum_{i=1}^N \phi(x_i, x'_{j_i^*}) \quad (2.4)$$

where

$$j_i^* = \operatorname{argmin}_{j=1, \dots, N'} \|x_i - x'_j\| \quad (2.5)$$

we constrain ourselves to matching points that are similar in the sense that they are close together (we can in fact use a metric in an induced space [5]). However, this technique is implicitly dependent on the choice of the origin, and the result of averaging dot products is difficult to interpret from the perspective of spatial similarity of a set of vectors (Figure 2.1). Therefore, the minor kernel itself must act as a similarity measure in equation (2.3). Choosing a radially symmetric kernel that is monotonically decreasing as $\|x_i - x'_j\|$ results in equations (2.3) and (2.4) being equivalent. The result of its evaluation can be thought of as the likelihood that the two vectors match each other in that space.

A special case where a simple dot product is appropriate is the case where the data are normalized. In this case, the data are constrained to lie on a hypersphere

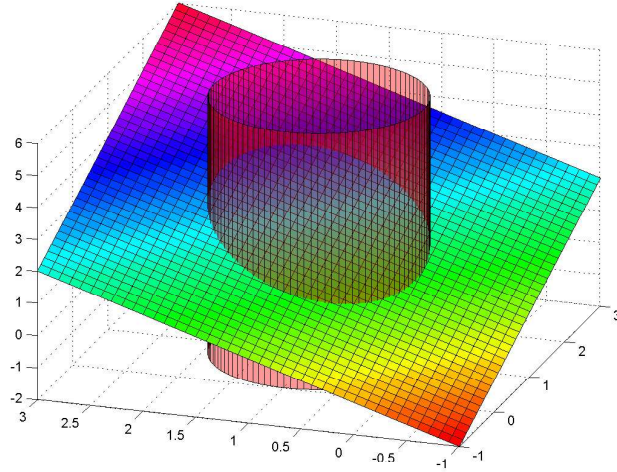


Figure 2.1. $\langle \{1, 1\}^T, \{x, y\}^T \rangle$ is plotted along with an iso-curve of $\|\{1, 1\}^T - \{x, y\}^T\|$, which result in a plane and a cylinder respectively. The averaging operation in equation (2.4) is difficult to interpret as we would expect to have equal contribution along the iso-curve. The equation is not coordinate free as changing the origin will result in different relative values for points selected by the argmin operation in equation (2.5)

and the dot product is in fact the cosine of the angle between two vectors. The cosine of the angle is radially symmetric with respect to a point on the manifold, and we in fact arrive at a radially symmetric, monotonically decreasing similarity metric, though we have no control over the scale at which we compare the data. Without such a geometric constraint, we must take greater care with our choice of kernel.

To explicitly interpret the minor kernel as representing the likelihood of a match between vectors, the kernel takes the shape of a density over the space. If we rely on the feature space, or a transformation of that space, to separate the data, then we require that the kernel have density inversely proportional to a monotonic function of a distance metric in the space. A Gaussian RBF kernel is in fact to a constant factor a density over the original feature space that has density inversely proportional to a monotonic function of distance.

2.5 Pyramid Match Kernel

Another recently proposed kernel between sets of vectors is the Pyramid Match Kernel [17]. Barla et al. first proposed histogram overlap as a kernel between images [1]. Grauman and Darrell extend this idea to a weighted multi-resolution histogram intersect measure over local features [17]. In order to have a fixed base scale for the histogram, they assume that the data are first scaled so that there is a minimum inter-vector distance of 1, and that the data are bounded by a sphere of known radius¹, r . Beginning with a bin width of $\frac{1}{2}$, each subsequent level in the pyramidal histogram has a bin width double the length of the previous and has half the number of bins. The kernel itself is defined to be

$$k(I, I') = \frac{\hat{k}(I, I')}{\sqrt{\hat{k}(I, I) \cdot \hat{k}(I', I')}} \quad (2.6)$$

$$\hat{k}(I, I') = \sum_{i=0}^{\lceil \log 2r \rceil} \alpha_i (|H_{I,i} \cap H_{I',i}| - |H_{I,i-1} \cap H_{I',i-1}|) \quad (2.7)$$

where α_i are weights for each histogram resolution, and $|H_{I,i} \cap H_{I',i}|$ is the histogram intersect between the images calculated with a bin width of 2^i ,

$$|H_{I,i} \cap H_{I',i}| = \sum_m \min(H_{I,i}^{(m)}, H_{I',i}^{(m)}) \quad (2.8)$$

At the base level, every feature falls into its own bin due to the bound on inter vector distance, and at level $\lceil \log 2r \rceil$, all of the features in each image fall into just one bin.

The setting for the weights

$$\alpha_i = \frac{1}{2^i} \quad (2.9)$$

¹Note that SIFT features are typically computed with integer arithmetic in order to trade quantization error for speed, giving a bound on the inter vector distance. Additionally, they are normalized and therefore lie on a hypersphere of known radius [30].

was proposed by Grauman and Darrell, though other weightings that are decreasing as i may be appropriate. At each level, a certain number of features in the respective images are close enough to fall into the same bin. The features that are close enough in bin i but not close enough in bin $i - 1$ each contribute α_i to the similarity measure. The previous matches are subtracted so that features that have already been matched are not counted multiple times. Additionally, the kernel is normalized by the self similarity of the images in order to avoid favoring images with a larger number of features.

Each of the kernels described in this chapter have some intuitive basis, but it is unclear where they are similar to one another and where they differ, and if there is an overarching justification for choosing one over another. The next section introduces a probabilistic framework for understanding the kernels with respect to the feature space.

CHAPTER 3

A PROBABILISTIC FRAMEWORK

Kernels between sets of vectors indicate a degree of similarity between two point clouds. This chapter describes a probabilistic framework for describing the similarity of point clouds based on the estimation of an underlying distribution for each set of vectors. This estimation is done non-parametrically, and the final kernel is computable in closed form.

3.1 Probability Product Kernels

A general approach for generating a kernel between distributions over observations was outlined in [22]. They propose *probability product* kernels of the form

$$k(p, p') = \int p(x)^\rho p'(x)^\rho dx \quad (3.1)$$

where p and p' are distributions that represent the two objects and ρ is a parameter of the family of kernels. There are two special cases of interest for ρ : Bhattacharyya's affinity between distributions [2]

$$k(p, p') = \int \sqrt{p(x)}\sqrt{p'(x)}dx \quad (3.2)$$

and the expected likelihood kernel

$$k(p, p') = \int p(x)p'(x)dx = E_p[p'(x)] = E_{p'}[p(x)] \quad (3.3)$$

While Bhattacharyya’s affinity between distributions is equal to one when the two distributions are identical, the expected likelihood kernel is unbounded for equal distributions, and favors distributions with low entropy.

Jebara, Kondor, and Howard derive closed form expressions for many parametric forms for $p(x)$ [22]. Of particular interest is that of the Gaussian distribution:

$$\int_{\mathbb{R}^D} p(x)^\rho p'(x)^\rho dx = \frac{1}{((2\pi)^{(2\rho-1)\rho})^{D/2}} \frac{|\Sigma^\dagger|^{1/2}}{|\Sigma|^{\rho/2} |\Sigma'|^{\rho/2}} e^{-\frac{\rho}{2}(\mu^T \Sigma^{-1} \mu + \mu'^T \Sigma'^{-1} \mu' - \mu^\dagger T \Sigma^\dagger \mu^\dagger)} \quad (3.4)$$

where

$$\Sigma^\dagger = (\Sigma^{-1} + \Sigma'^{-1})^{-1} \quad (3.5)$$

and

$$\mu^\dagger = \Sigma^{-1} \mu + \Sigma'^{-1} \mu' \quad (3.6)$$

The expected likelihood kernel applied to two spherical Gaussians of equal variance is equal to

$$k(p, p') = \frac{1}{(4\pi\sigma^2)^{D/2}} e^{-\|\mu' - \mu\|^2 / (4\sigma^2)} \quad (3.7)$$

which [22] point out is in fact equivalent to the Gaussian RBF kernel to a constant factor. Although one might argue that the expressive capacity of single Gaussians is overly restrictive¹, we will see in the following sections that the kernel between Gaussians is important in the derivation of more sophisticated results.

3.2 Kernel Density Estimation

Because our setting consists of images represented as sets of vectors, we need to estimate a distribution over the space in which the vectors lie. Kernel density

¹In fact, for the more restrictive formulation in equation (3.7) we have no more information than were we to represent the data by their centroid.

estimation estimates distributions from sampled data without assuming a parametric form [13, 42]. To estimate a density at a specific point, x , we use the formula

$$p(x) = \frac{k}{dV \cdot N} \quad (3.8)$$

where dV is a volume around the point, k is the number of samples that fall into the volume, and N is the number of points sampled. A window function, ϕ , is chosen to precisely define the region around x from which to construct the volume, and a scale parameter h is introduced to vary the width of the window. The number of samples, k , that falls into a region defined by this window is

$$k = \sum_{i=1}^N \phi\left(\frac{x - x_i}{h}\right) \quad (3.9)$$

We restrict ϕ to be normalized and non-negative, i.e. a distribution. Substitution into equation (3.8) yields

$$p(x) = \frac{1}{N} \sum_{i=1}^N \frac{1}{dV} \phi\left(\frac{x - x_i}{h}\right) \quad (3.10)$$

An interpretation of dV is that it represents the scaling factor that normalizes the window dependent on the dimensionality of the space:

$$\int \frac{1}{dV} \phi\left(\frac{x - x_i}{h}\right) dx = \int \phi(u) du = 1 \quad (3.11)$$

We will assume that our window functions, ϕ , are normalized so we will not explicitly account for $\frac{1}{dV}$ in subsequent equations. Of particular interest for our purposes is the Gaussian kernel, defined by²

²We slightly abuse notation here by treating ϕ as a function of one variable above (as in the kernel density estimation literature), and as a function of two variables below. In practice, this

$$\phi(x_i, x) = \frac{1}{(2\pi\sigma^2)^{D/2}} e^{-\|x_i - x\|^2/2\sigma^2} \quad (3.12)$$

3.2.1 Probability Product Kernels Applied to Kernel Density Estimates

We now consider what happens when we use the probability product kernel to compare two distributions that have been estimated from samples using kernel density estimation:

$$k(p, p') = \int \left(\frac{1}{N} \sum_{i=1}^N \phi(x_i, x) \right) \cdot \left(\frac{1}{N'} \sum_{j=1}^{N'} \phi(x'_j, x) \right) dx \quad (3.13)$$

where $\phi(x_i, x)$ is given as in equation (3.12). Rearranging terms in equation 3.13 we arrive at

$$k(p, p') = \frac{1}{N} \frac{1}{N'} \sum_{i=1}^N \sum_{j=1}^{N'} \int \phi(x_i, x) \cdot \phi(x'_j, x) dx \quad (3.14)$$

Since the inner integral is simply the expected likelihood kernel between Gaussians, the end result is

$$k(p, p') = \frac{1}{N} \frac{1}{N'} \frac{1}{(4\pi\sigma^2)^{D/2}} \sum_{i=1}^N \sum_{j=1}^{N'} e^{-\|x'_j - x_i\|^2/(4\sigma^2)} \quad (3.15)$$

when the Gaussian is isotropic and equal variance. Kernels of this form were an intermediate step in the development of multi-instance kernels [15] but without a probabilistic justification. More recently, this form was suggested for mixtures of exponential distributions of which Gaussian kernel density estimation is a special case [21].

Bhattacharyya's affinity between Gaussians has intuitive appeal due to that it is equal to 1 when the two distributions are identical, and equal to 0 when there is no

makes no difference when dealing with the Gaussian kernel because it, as above, is a function of the difference between two points. From now on, ϕ represents a function of two variables as in the kernel literature.

overlapping support. As such, we expect that we would be less likely to encounter unexpected behavior in the kernel in the case where we have low entropy distributions, which can return very high kernel evaluations. Unfortunately, a closed form expression is not apparent in the case that p and p' are mixtures. An approximation for mixtures, which when applied to kernel density estimation yields

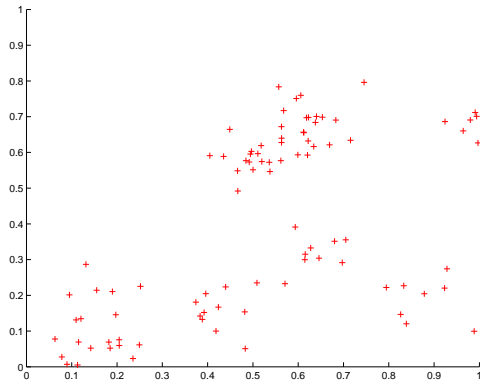
$$\int \left(\frac{1}{N} \sum_{i=1}^N \phi(x_i, x) \right)^{\frac{1}{2}} \cdot \left(\frac{1}{N'} \sum_{j=1}^{N'} \phi(x'_j, x) \right)^{\frac{1}{2}} dx \approx \frac{1}{N} \frac{1}{N'} \sum_{i=1}^N \sum_{j=1}^{N'} \int \sqrt{\phi(x_i, x)} \sqrt{\phi(x'_j, x)} dx \quad (3.16)$$

was proposed in [21], while Gibbs sampling was suggested in [22]. The heuristic solution does not have theoretical rigor, while Gibbs sampling is undesirable, especially in high dimensions. We further analyze the expected likelihood of kernel density estimations in the following sections.

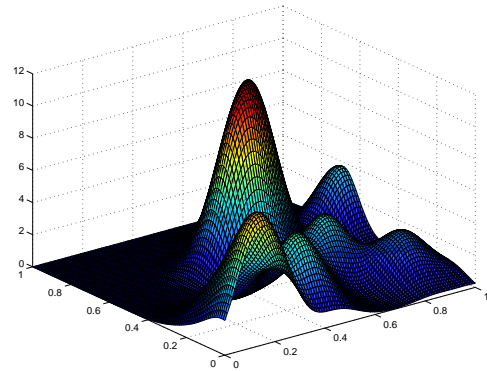
3.2.1.1 Example in Two-Dimensions

Figure 3.1 outlines the probability product kernel applied to two-dimensional data. In Figure 3.1(a) a set of vectors is represented by its scatter plot. The vectors were generated from a mixture of 10 Gaussians with means drawn from a uniform distribution over the unit square, and $\sigma = 0.05$. Figure 3.1(b) shows the estimated density over that same set of vectors. Figures 3.1(c) and 3.1(d) show densities estimated from samples of the same mixture and of a different mixture, respectively. Finally, Figures 3.1(e) and 3.1(f) show the product of these two densities with the original. As the densities estimated from two different sets of samples from the same distribution have higher overlap than the two densities estimated from samples of different distributions, the integral of the probability product will be much higher for densities estimated from the same underlying distribution.

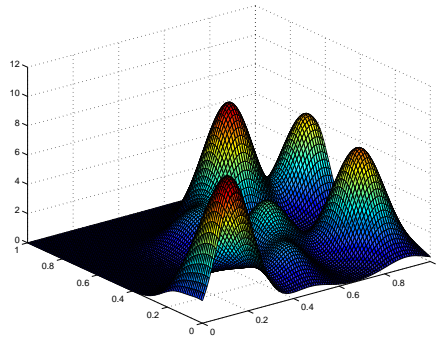
For simplicity, we continue with the assumption that the kernel is isotropic. This in fact does not turn out to be a large setback as the space itself can be transformed



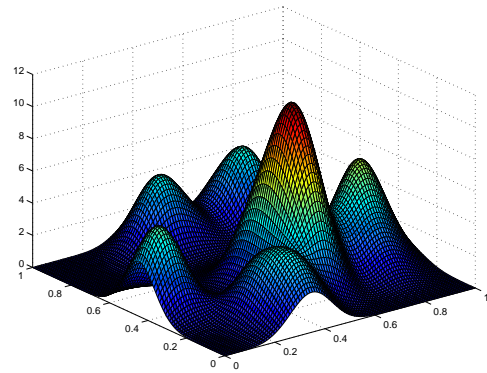
(a) A vector set



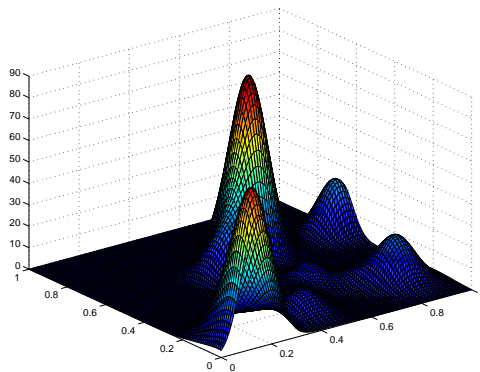
(b) Density estimated from set of vectors



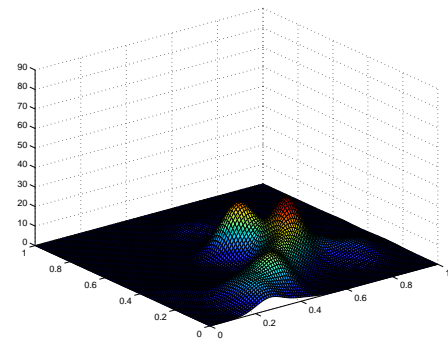
(c) Estimated density from a different sample of the same class



(d) Estimated density from a sample of a different class



(e) Probability product of the densities shown in figures 3.1(b) and 3.1(c)



(f) Probability product of the densities shown in figures 3.1(b) and 3.1(d)

Figure 3.1. Probability product applied to density estimations of two-dimensional data

prior to classification [16]. Alternatively we could use the more general formulation of the kernel between Gaussians in [22] that places no restrictions on the covariance matrices (equation (3.4)).

3.2.2 Bandwidth Selection

In choosing the parameter, σ , there are several sensible approaches one could consider. A common technique applied to density estimation is to leave out each single point one at a time and maximize the likelihood of that point with respect to all the other points

$$\sigma^* = \operatorname{argmax}_{\sigma} \sum_{i=1}^m \log \left(\frac{1}{n-1} \sum_{j \neq i} \phi_{\sigma}(x_i, x_j) \right) \quad (3.17)$$

where ϕ_{σ} is the window function calculated at scale σ . This does not exactly make sense in the case where we are learning over sets of vectors, however, as we are maximizing the likelihood of only individual points with respect to other points. This does not directly say anything about the likelihood of the set of points as a whole.

We could instead leave out each single collection of points and maximize the likelihood of that set of points with respect to all the other points

$$\sigma^* = \operatorname{argmax}_{\sigma} \sum_{i=1}^n \log \left(\frac{1}{n-1} \sum_{j \neq i} E_{p_{i,\sigma}}(p_{j,\sigma}) \right) \quad (3.18)$$

where $p_{i,\sigma}$ and $p_{j,\sigma}$ are density estimates at scale σ and the expectation is calculated as in equation (3.15). This solution treats the problem as an estimation of density over a set of distributions, where each distribution represents an entire image.

A discriminative approach may yield better results with respect to classification. Recent work on estimating discriminative densities includes [32]. More directly, we could employ a wrapper technique to determine bandwidth directly from the performance of the classifier on a test set of images.

3.3 Similarities Between the Expected Likelihood Kernel with Density Estimation and Previous Approaches

The expected likelihood kernel between kernel density estimations has a very similar form to the matching kernel with a Gaussian RBF as the minor kernel (equations (2.2) and (3.15)). Aside from a constant factor, the only difference is that the matching kernel sums the contribution only from the closest match via the max operation, while the expected likelihood kernel between kernel density estimations sums over every contribution. Density estimation uses a lower variance statistic than the matching kernel and there are no discontinuities introduced as a result of the max operation.

The pyramid match kernel (equation (2.6)) itself can be viewed as an approximation to the matching kernel, and therefore an approximation to the estimate of the expectation of one distribution (which generated one set of points) with respect to another distribution (which generated the second set of points). At each level of the pyramid, the distance between the newly matched vectors is constrained to be somewhere between 2^{i-1} and $\sqrt{D} \cdot 2^i$, where D is the number of dimensions of the feature space. If we know a density for the features over the manifold in which they lie, or if we assume a uniform density, we can estimate the expected distance, d_i , between newly matched features at each level. If we replace the assignments for the weights, α_i , given in equation (2.9) with an assignment that samples a Gaussian RBF kernel

$$\alpha_i = e^{-d_i^2/2\sigma^2} \tag{3.19}$$

the similarities between the two become apparent.

3.3.1 Robustness Issues

The similarity between the matching kernel and the expected likelihood kernel gives rise to the question of what statistics are appropriate to allow maximum dis-

crimination while maintaining robustness. Robust estimators are those that make use of some subset or weighting of the data to reduce the effect of outliers [18, 19]. One of the most simple techniques for selecting a subset of the data is via order statistics. By including only a certain quantile of data, outliers will fall in the excluded range and the estimator will not be effected. The max operation in the matching kernel is in fact an order statistic that excludes every data point except the closest match. This is a reasonable choice in the event that it is assumed that each local feature in one image matches exactly one feature in the other. Alternative statistics would include averaging over quantiles of the ordered data, or hybrid approaches in which the tradeoff between an estimator based on all the data, or on just a portion of the data, are controlled by a parameter of the estimator [19].

In terms of discrimination, we wish to select an estimator that is robust to values that give little or misleading information about class membership. In general, we wish to choose a statistic that gives a higher similarity value to objects of the same class and a lower similarity value to objects of different classes. This is a data dependent choice, and without further assumptions about the distribution from which the data are drawn, the statistic used must be chosen experimentally.

There is in fact a certain amount of robustness built into any system that calculates statistics over Gaussian kernel evaluations. Because

$$\lim_{|x_i - x'_j| \rightarrow \infty} \phi(x_i, x'_j) = 0 \quad (3.20)$$

outliers will tend to have a limited effect on the summation. However, experimental results show that the choice of estimator does have a significant effect on classification performance.

In this chapter we have presented a probabilistic framework for non-parametrically estimating the similarity between sets of vectors using kernel density estimation with probability product kernels. This has lead to the development of the expected like-

likelihood kernel (equation (3.15)). Interestingly the pyramid match kernel with appropriate weighting is an approximation to the matching kernel with a Gaussian minor kernel, and both of these kernels can be better understood by their relation to the expected likelihood kernel.

CHAPTER 4

COMBINING MULTIPLE SOURCES OF INFORMATION

Ensemble methods are learning algorithms that have been shown to improve performance by combining the outputs of multiple component classifiers. Ensemble methods for classification have been shown to have better accuracy than the component classifiers if the component classifiers are accurate and diverse [9]. An accurate classifier is one that outperforms random guessing, and diverse classifiers are those that produce independent errors. Typically, application of ensemble methods focuses largely on inducing independence of errors by manipulating the training set, manipulating the input features, or injecting randomness in the learning algorithm.

Despite the advantages of local features, global features are still useful in applications where a rough segmentation of the object of interest is available. Due to the fundamental difference in how local and global features are computed, we expect that the two representations would provide different kinds of information. Most local features represent texture in an image patch (c.f. Section 1.1). Global features include contour representations, shape descriptors, and texture features. Global texture features and local features provide different information about the image because the support over which texture is computed varies. We expect classifiers that use global features will commit errors that differ from those of classifiers based on local features [28].

We certainly have a degree of independence between local and global features, so a classification system that made use of both kinds of features would likely perform much better than a classifier based on either type alone. One could combine the out-

puts of several base classifiers trained on either local or global classifiers using a fixed strategy, or a meta-learning technique such as stacking [41, 28]. Alternatively, one could design a kernel that combines the discriminative abilities of kernels computed over different feature types. This is a general technique that extends to multiple view learning in general, e.g. classification of video segments using both audio and video channels simultaneously [36].

4.1 Multiple View Kernels

It is common practice to combine multiple sources of information in a single kernel. For example, when the matching kernel was first proposed [45], a variant was suggested that made use of the relative positions of the local features in the image space. This variant replaced the minor kernel in equation (2.3) with

$$\phi'(x_i, x'_j) = \phi(x_i, x'_j) \cdot e^{-(l(x_i) - l(x'_j))^2 / 2\sigma^2} \quad (4.1)$$

where $l(x_i)$ is the coordinate of the feature in the image. The authors sought to constrain the kernel to match objects that have components with similar appearance as well as similar spatial layout within the image, and they did so by *multiplying* the minor kernel with a Gaussian RBF kernel between alternate representations of the local features, namely their x and y coordinates. What technique for combining kernels for different object representations is the most effective for improving classification accuracy? To begin to answer this question, let us consider two techniques for combining kernels, multiplication (as above) and addition.

To understand the effects of adding two kernels, consider the interpretation in embedding space. Assume two positive definite kernels

$$k_1(x_1, x_2) = \langle \Phi_1(x_1), \Phi_1(x_2) \rangle \quad (4.2)$$

$$k_2(x_1, x_2) = \langle \Phi_2(x_1), \Phi_2(x_2) \rangle \quad (4.3)$$

The sum of the two is

$$k_{sum}(x_1, x_2) = k_1(x_1, x_2) + k_2(x_1, x_2) \quad (4.4)$$

$$= \langle \Phi_1(x_1), \Phi_1(x_2) \rangle + \langle \Phi_2(x_1), \Phi_2(x_2) \rangle \quad (4.5)$$

$$= \left(\sum_{i=1}^{|\Phi_1(x)|} \Phi_1(x_1)^{(i)} \cdot \Phi_1(x_2)^{(i)} \right) + \left(\sum_{j=1}^{|\Phi_2(x)|} \Phi_2(x_1)^{(j)} \cdot \Phi_2(x_2)^{(j)} \right) \quad (4.6)$$

where $\Phi_1(x_1)^{(i)}$ is the i th element of the vector $\Phi_1(x_1)$ in the embedding space. This is itself equivalent to a dot product in an induced space of dimensionality $|\Phi_1(x)| + |\Phi_2(x)|$. One can see that data that are separable in either of the two embedding spaces are separable in the augmented space. To the degree that the individual kernels are independent, this will increase overall separability and therefore classification accuracy. If we wish to control the bias accorded to each of the component kernels, we can weigh them differently

$$k_{sum}(x_1, x_2) = \alpha k_1(x_1, x_2) + (1 - \alpha) k_2(x_1, x_2) \quad (4.7)$$

where $0 \leq \alpha \leq 1$.

To understand the effects of multiplying two kernels, consider again the interpretation in embedding space.

$$k_{prod}(x_1, x_2) = k_1(x_1, x_2) \cdot k_2(x_1, x_2) \quad (4.8)$$

$$= \langle \Phi_1(x_1), \Phi_1(x_2) \rangle \cdot \langle \Phi_2(x_1), \Phi_2(x_2) \rangle \quad (4.9)$$

$$= \left(\sum_{i=1}^{|\Phi_1(x)|} \Phi_1(x_1)^{(i)} \cdot \Phi_1(x_2)^{(i)} \right) \left(\sum_{j=1}^{|\Phi_2(x)|} \Phi_2(x_1)^{(j)} \cdot \Phi_2(x_2)^{(j)} \right) \quad (4.10)$$

$$= \sum_{i=1}^{|\Phi_1(x)|} \sum_{j=1}^{|\Phi_2(x)|} (\Phi_1(x_1)^{(i)} \cdot \Phi_2(x_1)^{(j)}) (\Phi_1(x_2)^{(i)} \cdot \Phi_2(x_2)^{(j)}) \quad (4.11)$$

We can perform a variable substitution where a new variable k ranges over i and j , and we see that the sum in equation (4.11) is itself equivalent to a dot product.

The dimensions of this transformed space consist of the products of every pairwise combination of dimensions in the embedding spaces of the two component kernels.

It is not clear if and when the product of two kernels would perform better than the sum, but the sum is attractive for its interpretation with respect to separability. It is of course possible to simultaneously capture the advantages of both techniques with minimal computation

$$k_{poly}(x_1, x_2) = (k_1(x_1, x_2) + 1) \cdot (k_2(x_1, x_2) + 1) \quad (4.12)$$

$$= k_1(x_1, x_2) \cdot k_2(x_1, x_2) + k_1(x_1, x_2) + k_2(x_1, x_2) + 1 \quad (4.13)$$

Any separability introduced by either the sum or the product of the original kernels will also be available in this kernel. One can see this by noting that equation (4.13) is the sum of equations (4.8) and (4.4) and then applying the analysis in equation (4.6). We can again use a parameter, α , to control the weight given to each term in equation (4.13)

$$k_{poly}(x_1, x_2) = (k_1(x_1, x_2) + (1 - \alpha)) \cdot (k_2(x_1, x_2) + \alpha) \quad (4.14)$$

We can recursively apply the analysis above to understand the embedding space of any multivariate polynomial with an arbitrary number of component kernels. However, we need to be careful about overfitting, which is quite easy to do by considering higher order terms.

CHAPTER 5

EXPERIMENTAL RESULTS





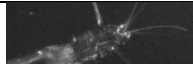
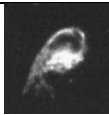





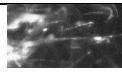
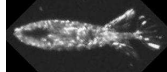

5.1 Video Plankton Recorder Data Set

One of the applications of these techniques is to marine science data collected by a tool called the Video Plankton Recorder (VPR) [8]. The Video Plankton Recorder captures images of multicellular organisms that have organs and appendages with distinct visual appearances (Figure 5.1). The data set consists of 1826 gray-scale images that belong to one of 14 classes (Table 5.1), which have been identified by experts [28]. The data set is challenging from a classification viewpoint for several reasons. Organisms are photographed from arbitrary three-dimensional views. The size of the organisms relative to the field of view of the camera results in many images in which the organism is only partially visible. It is consequently a challenging and attractive data source for testing our methods. The data set is described in greater detail in [28].

5.1.1 Comparison of Matching Kernel and Expected Likelihood Kernel

We report results here for a comparison of the matching kernel and the expected likelihood kernel (equation (3.15)). The first experiment consisted of comparing the accuracy estimate from ten-fold cross validation for a range of bandwidths (σ). Classifications were computed using the libsvm library [7]. Results are shown in Figure 5.2. We see that the matching kernel consistently outperforms the expected likelihood kernel for all bandwidths, and that the maximum accuracy for both the matching kernel and the expected likelihood kernel occurs at around $\sigma = 275$. The superior

Table 5.1. Taxonomic categories of images in the VPR data set

Category Name	Taxonomic Group	images	example
<i>Calanus finmarchicus</i>	copepod species	132	
Chaetognaths	zooplankton phylum	86	
<i>Conchoecia</i> Ostracods	ostracod genus	100	
Ctenophores	zooplankton phylum	34	
Euphausiids	zooplankton order	131	
Hyperiid Amphipods	zooplankton suborder	68	
Pteropods	zooplankton order	142	
Diatom Rods	phytoplankton class	97	
Larvaceans	zooplankton class	133	
Small Copepods	zooplankton class	433	
Unidentified Cladocerans	zooplankton order	108	
Siphonophores	zooplankton suborder	202	
<i>Euchaeta norvegica</i>	copepod species	81	
Siphonulae	developmental stage of zooplankton suborder	78	

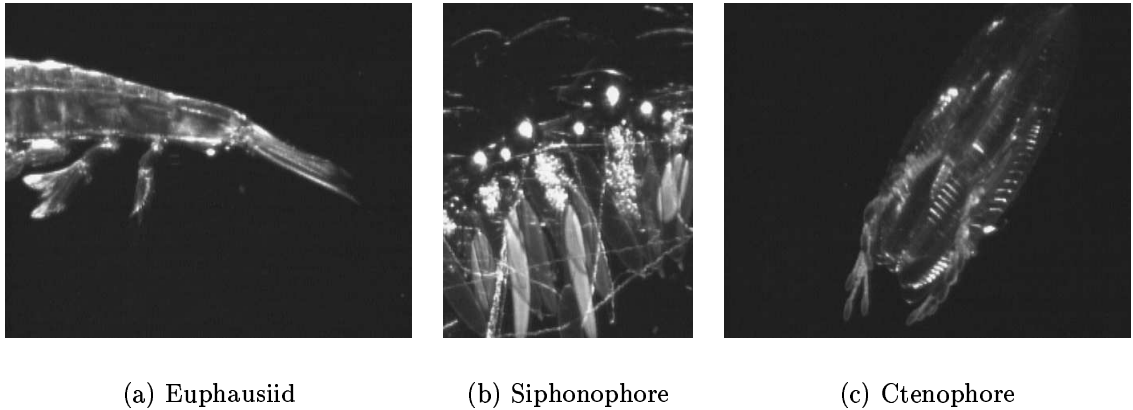


Figure 5.1. A few example images from the VPR data set.

performance of the matching kernel may be due to the fact that there are insufficient samples in any given image to robustly estimate a density. Because clusters of local features tend to have semantic coherence [12], we expect that the co-occurrence of samples from these clusters will form peaks in the estimated density. If there are insufficient samples in an image, peaks may not be robustly estimated and the process may be overwhelmed by spurious features.

In order to understand better the reasons why the matching kernel outperforms the expected likelihood kernel, we have run experiments that explore the distributions over distances between the individual features. The experiments consisted of selecting two example images from three different classes. We designate one of the images from each class as a query, and the other is designated a target. We only evaluate one sided distance in this case. The images used are shown in Table 5.2. Table 5.3 shows the histograms of distances between all pairs of points where one point is in the query image and the other is in the target image, while Table 5.4 shows the histograms of distances between all points in the query image, and the closest point in the target image. We can see that the expected likelihood kernel has to choose between distributions over distances that vary only very slightly, while the matching

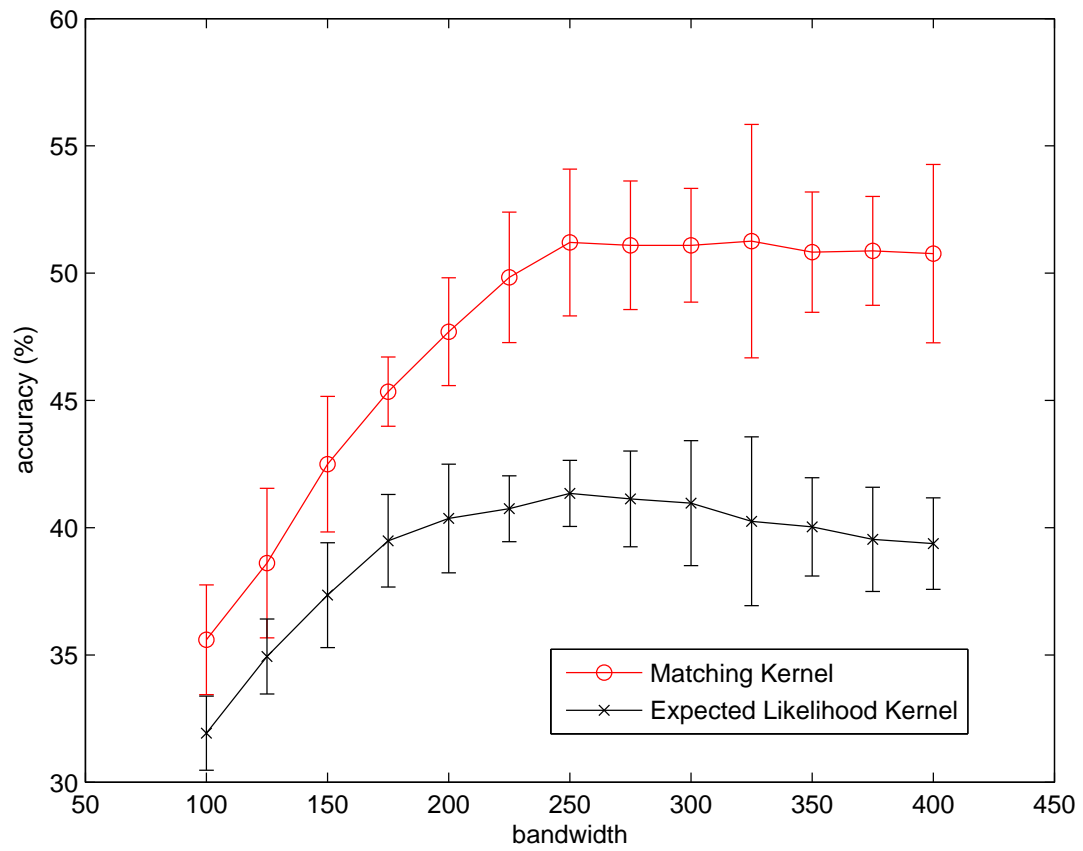
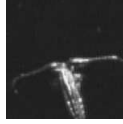
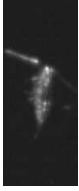






Figure 5.2. Bandwidth, σ , is plotted vs. accuracy on the VPR data set. Results are shown for the matching kernel, and for the expected likelihood kernel.

Table 5.2. VPR images used for distance experiments

Class	Query Image	Target Image
<i>Calanus finmarchicus</i>		
Chaetognaths		
<i>Conchoecia</i> Ostracods		

kernel has more variation in the distributions from a given query image. In fact, only very few features that closely match will have high values of the minor kernel in the average operation in the matching kernel (equation (2.3)).

5.1.2 Robustness Experiments

Because of the superior performance of the matching kernel over the expected likelihood kernel, we define a class of kernels parametrized by the order statistics as follows

$$k_{\beta}(I, I') = \frac{1}{2}[\hat{k}_{\beta}(I, I') + \hat{k}_{\beta}(I', I)] \quad (5.1)$$

$$\hat{k}_{\beta}(I, I') = \frac{1}{N} \frac{1}{\lceil \beta N' \rceil} \sum_{i=1}^N \sum_{\{x'_j | \phi(x_i, x'_j) \geq \phi(x_i, I'_{\beta})\}} \phi(x_i, x'_j) \quad (5.2)$$

where $\phi(x_i, I'_{\beta})$ is the $\lceil \beta N' \rceil$ th largest kernel evaluation ranging over the set of vectors, I' . β represents the fraction of kernel evaluations that will be averaged in the inner loop of the double summation. In the case that $\beta = \frac{1}{N'}$ this is equivalent to the matching kernel, and in the case that $\beta = 1$ the kernel becomes equivalent to the expected likelihood kernel.

For these experiments we held the bandwidth fixed at $\sigma = 275$ and varied the parameter β . Results are shown in Figure 5.3. We can see that we do in fact get an

Table 5.3. Histograms of distances over all pairwise feature matches. Each row represents a different query image, while the columns are each a different target image.

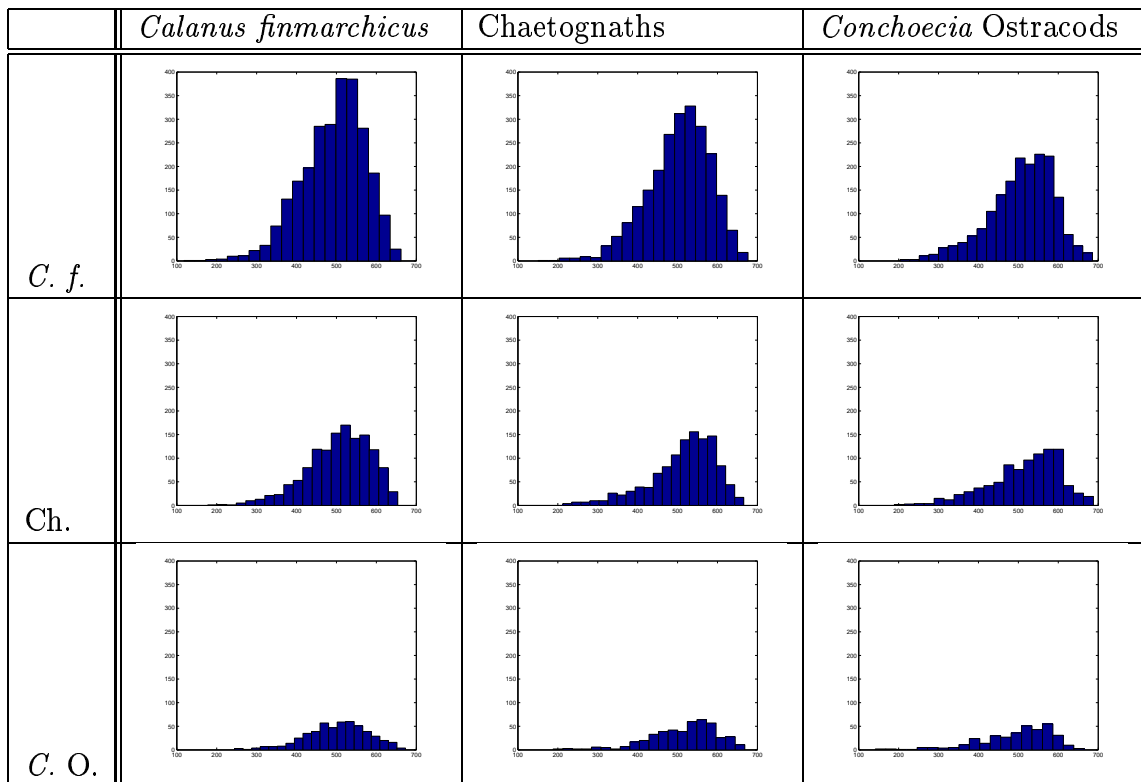
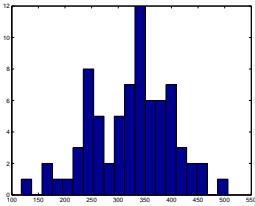
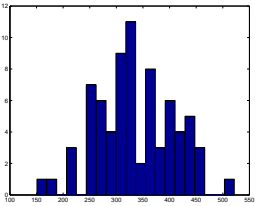
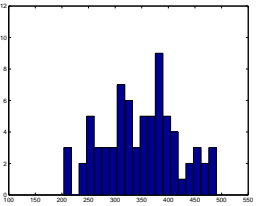
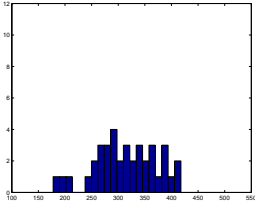
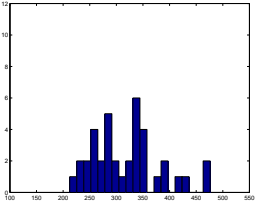
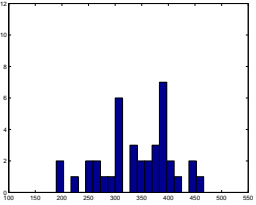
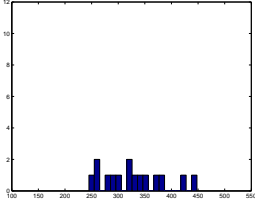
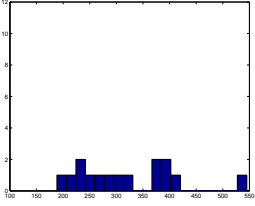
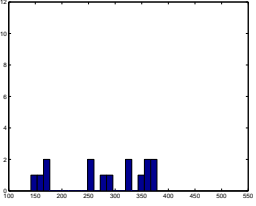


Table 5.4. Histograms of distances over maximum feature matches. Each row represents a different query image, while the columns are each a different target image.

	<i>Calanus finmarchicus</i>	Chaetognaths	<i>Conchoecia</i> Ostracods
<i>C. f.</i>			
Ch.			
<i>C. O.</i>			

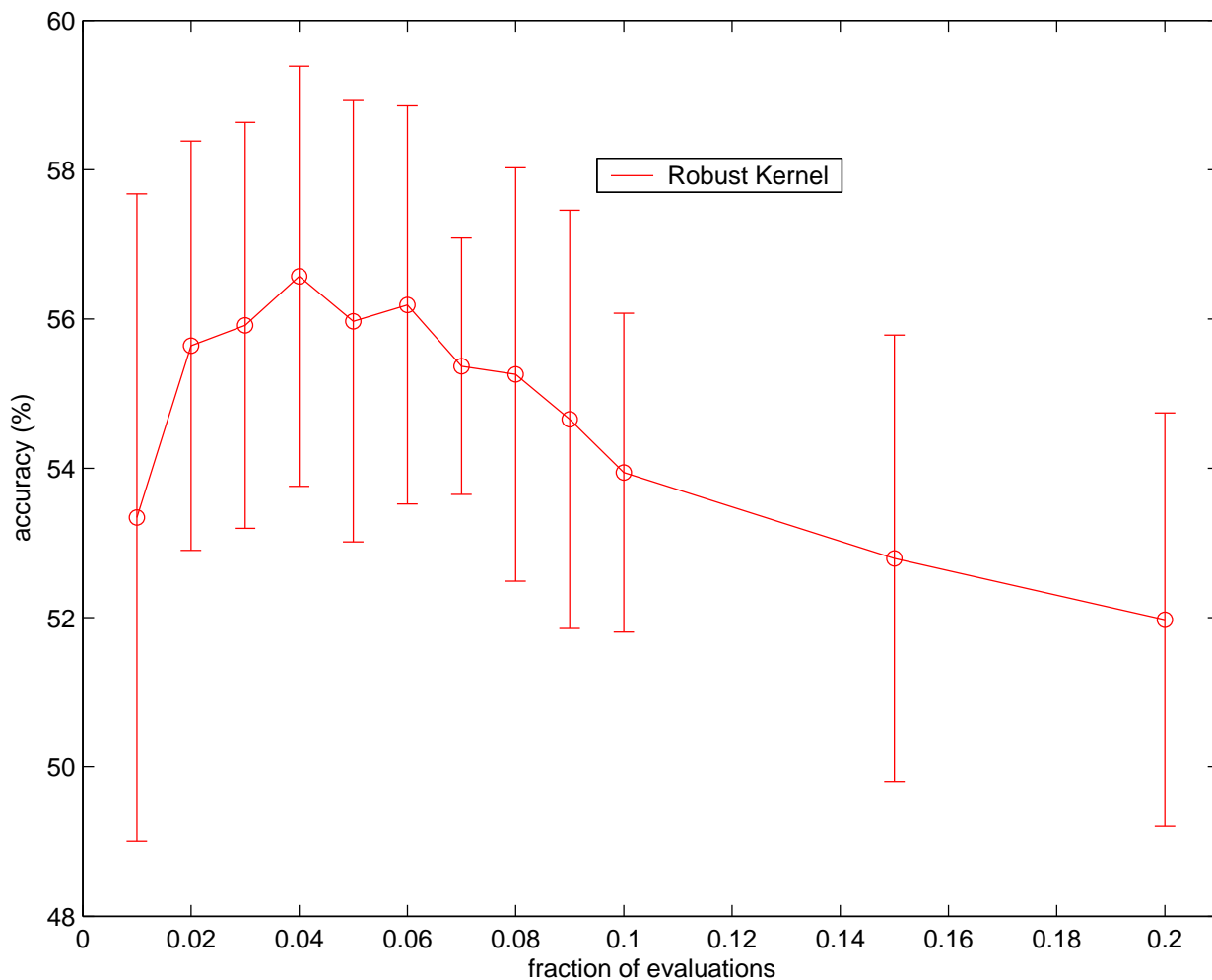


Figure 5.3. Fraction of minor kernel evaluations, β , is plotted vs. accuracy on the VPR data set. The kernel is computed as in equation 5.1.

increase in performance by averaging over a small fraction (approximately 5%) of the kernel evaluations as opposed to only the maximum match.

5.1.3 Combining Local and Global Features

Experiments have also been run using global features with the idea of combining global and local features for improved classification performance. Global features used here are those reported in [28]. Results for a global feature classifier using a Gaussian RBF kernel with varying bandwidths are reported in figure 5.4. The local

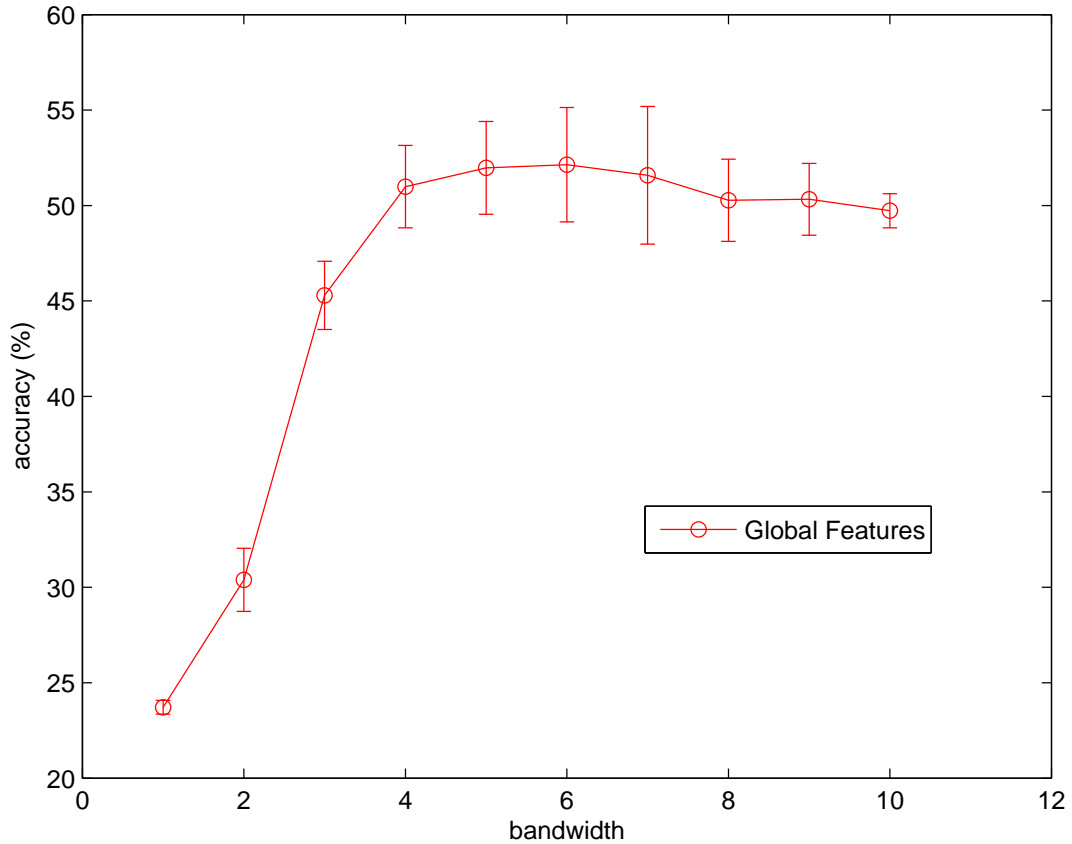


Figure 5.4. Bandwidth, σ , is plotted vs. accuracy on the VPR data set. Results are shown for global features using a Gaussian RBF kernel.

feature kernel used here is the matching kernel. There was no statistically significant improvement when the robust kernel was used in combination experiments.

5.1.3.1 Sum of Local and Global Kernels

When we combine local and global features using the mean of their kernels, a significant increase in accuracy results (figure 5.5). The accuracies show a plateau in the vicinity of the maximum, which gives us some robustness to suboptimal bandwidth selection. Interestingly, though, the maximum does not occur at the combination of the maximum matching kernel and the maximum global feature kernel. The peak

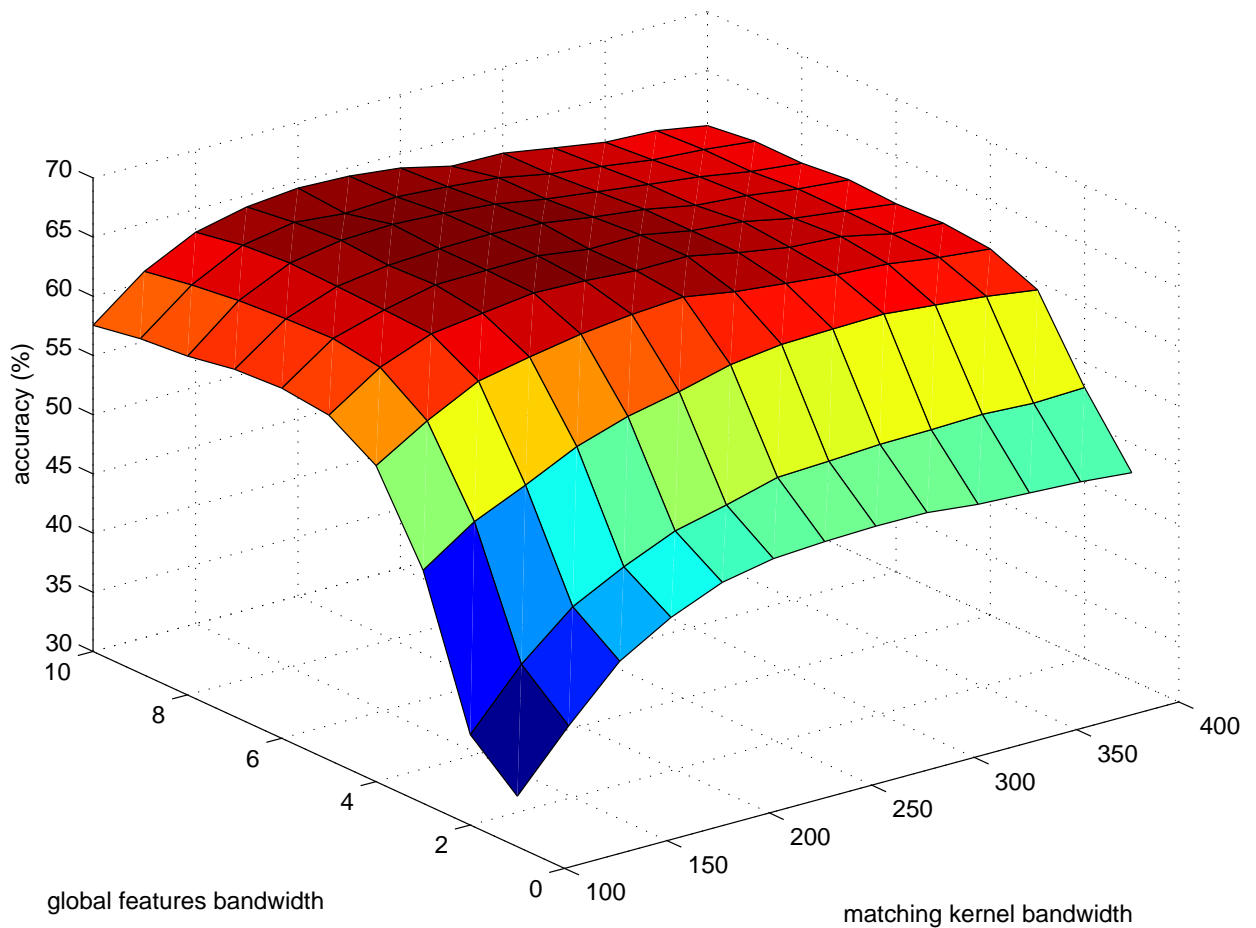


Figure 5.5. Classification accuracies for the mean of the kernels for global and local features.

instead is located at a point where the matching kernel bandwidth is somewhat smaller than that indicated in figure 5.2, and at a point where the global feature kernel bandwidth is somewhat larger than that indicated in figure 5.4. In an ensemble, the accuracy of the classifier is dependent on that errors between components are independent, rather than that the accuracy of each component is maximized.

Fixing the local bandwidth at 7 and the global bandwidth at 200, we have run experiments in which the weight accorded to global and local kernels varies as in equation 4.7. Results are shown in Figure 5.6. We can see that the performance varies

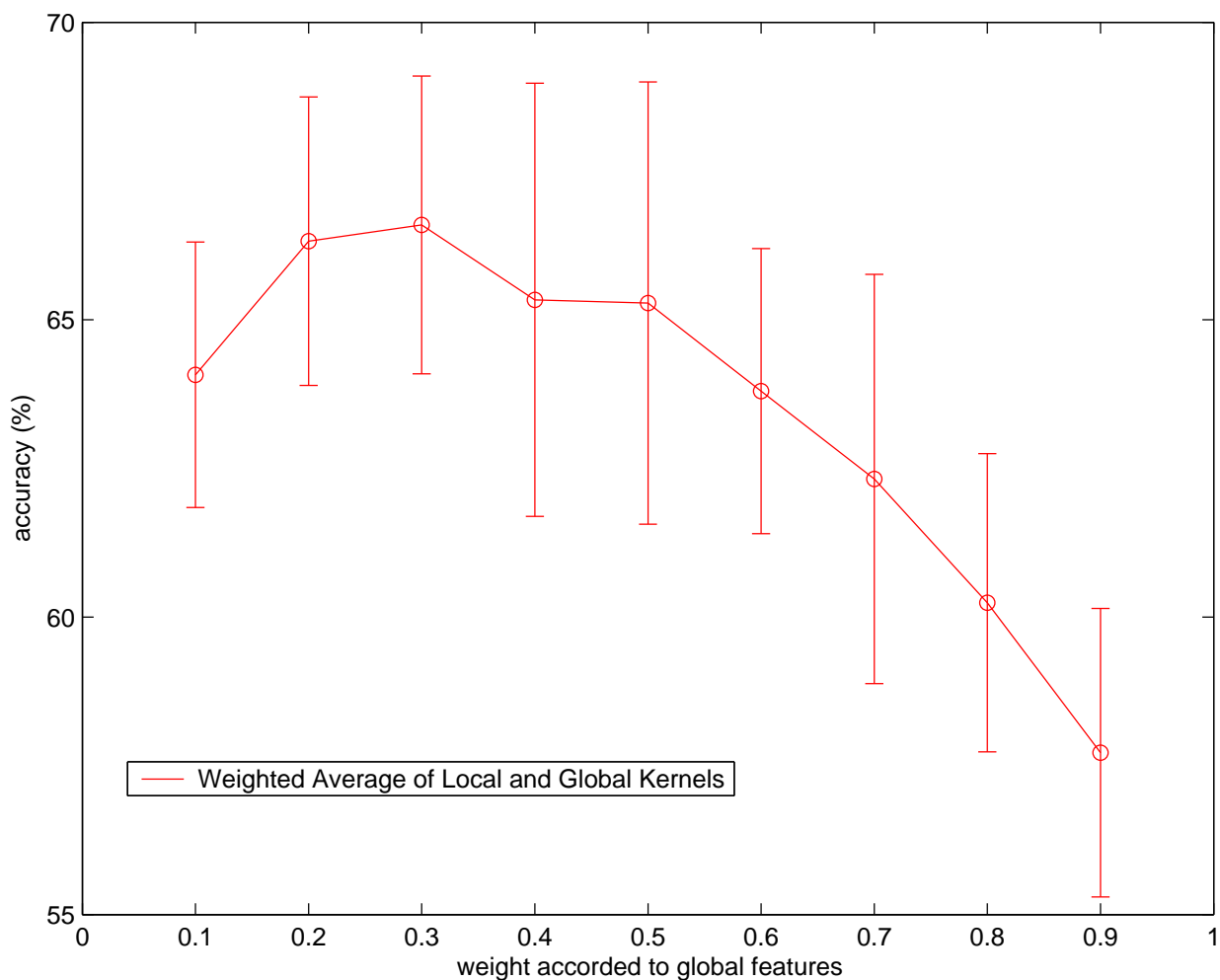


Figure 5.6. Classification accuracies for weighted averages of the kernels for global and local features.

as the parameter α changes with a maximum performance when global features are given a weight of approximately 0.3 and local features a weight of 0.7.

5.1.3.2 Product of Local and Global Kernels

Results are also shown in Figure 5.7 for the product of the component kernels. Accuracies are similar between the product and the mean, though the product performs slightly better for a larger global feature bandwidth. The difference between the best

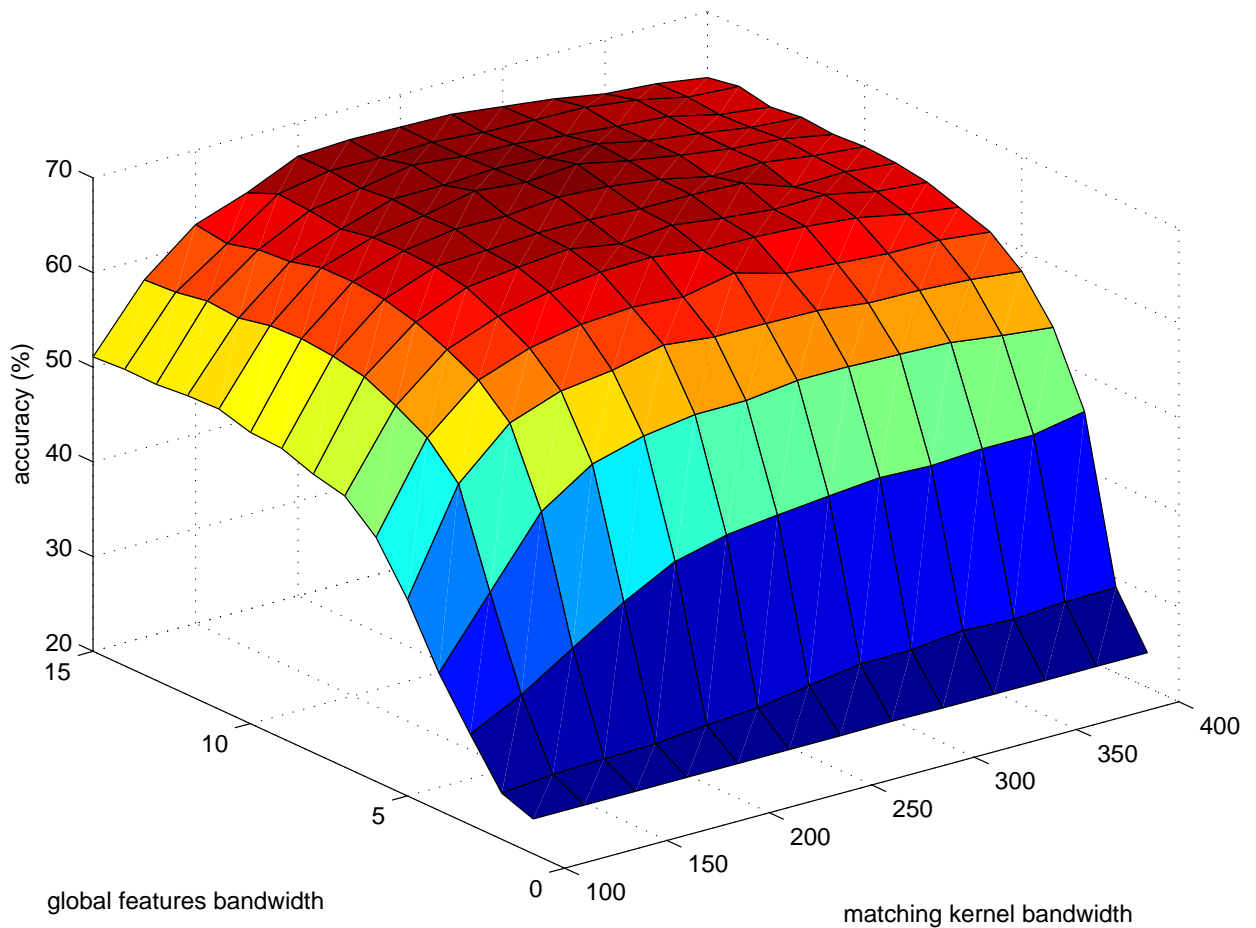


Figure 5.7. Classification accuracies for the product of the kernels for global and local features.

performance of the mean and the best performance for the product is statistically insignificant, $65.1 \pm 2.7\%$ vs $67.4 \pm 3.9\%$, respectively.

5.1.3.3 Polynomial Combination of Local and Global Kernels

For these experiments, we ran a simple polynomial combination of local and global features as in equation 4.14. In the first experiment, we used an equal weighting between the kernels computed for local and global features, shown in Figure 5.8. The maximum accuracy of approximately 72% is the best accuracy achieved thus far on the VPR data set. Additionally, we can see in Figure 5.8 that there is a relatively

broad plateau in the graph of the accuracy, indicating that the kernel allows for a good amount of tolerance to different settings of the parameters without significant loss of classification accuracy.

In the second experiment, we varied the weight of the local and global kernels. Due to the high performance of the weighting in Section 5.1.3.1 we set the kernel over global features to have a weight of 0.3 and the kernel over local features a weight of 0.7. Results for this weighting are shown in figure 5.9. The weighting performed slightly worse than the unweighted experiment described in the previous paragraph. This indicates that the weighting may not add significantly to the discriminability of the kernel.

5.2 ETH-80 Data Set

Comparative studies [14, 17] have used the ETH-80 data set as a baseline [27]. The ETH-80 data set consists of 8 different classes with 41 different views of 10 different examples per class. Example images are shown in figure 5.10. Results are reported here for several variations of local kernels to aid in comparison with other techniques that have results reported for the same data set. In our experiments here, we used ten-fold cross-validation with all images of one object from each class held out in each fold.

Results are shown for the matching kernel and expected likelihood kernel applied to the ETH-80 data set in figure 5.11. We can see that the matching kernel again outperforms the expected likelihood kernel by quite a large margin. The results of the matching kernel are consistent with those reported in [14].

Robustness experiments have been run on the ETH-80 data set as described in Section 5.1.2. Again, the bandwidth was held constant, $\sigma = 275$, while the fraction of evaluations included in the kernel, β , was allowed to vary. Results are shown in figure 5.12. We can see that unlike the VPR data set, there is no advantage from

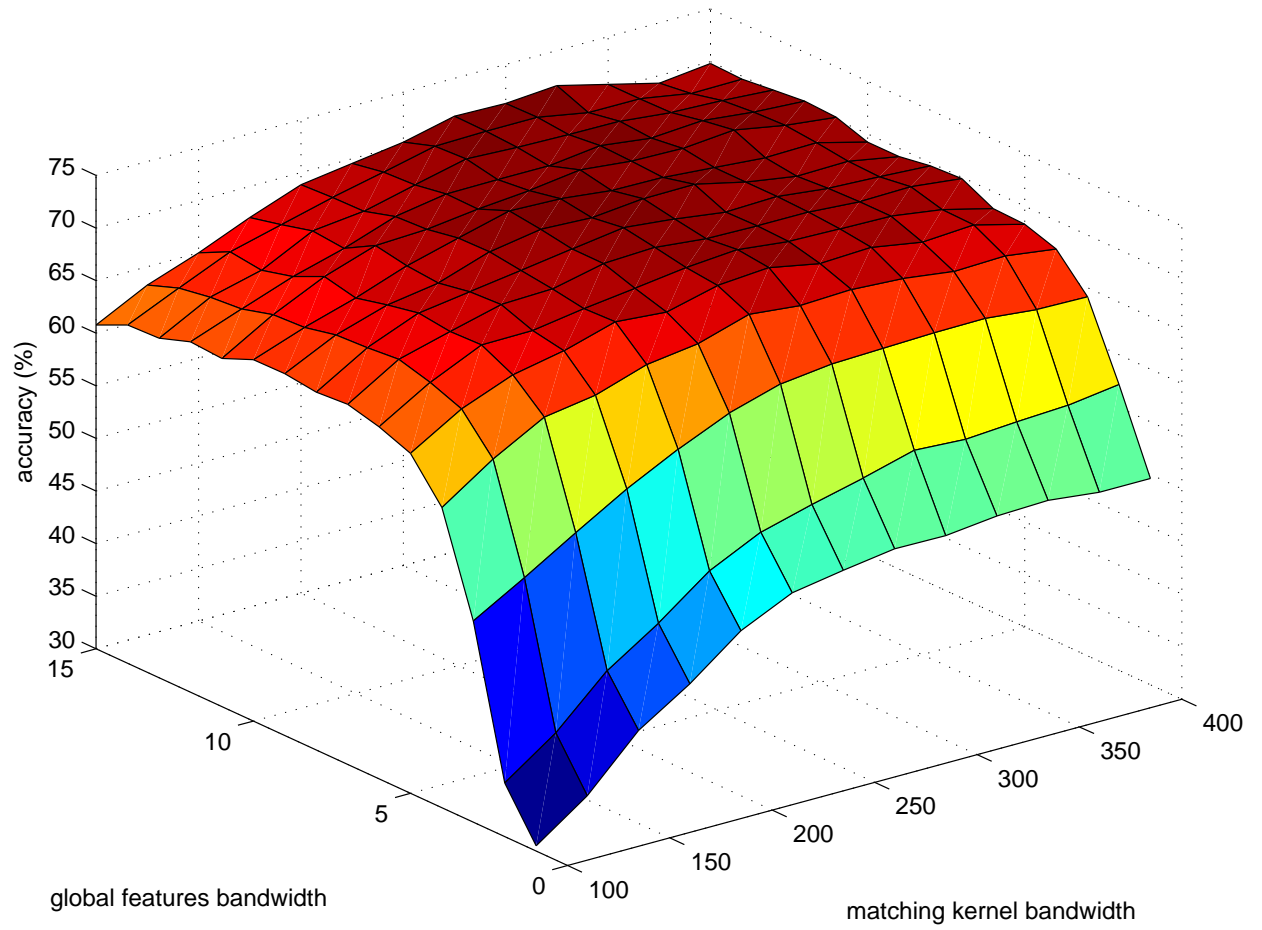


Figure 5.8. Classification accuracies for an unweighted polynomial combination of the kernels for global and local features.

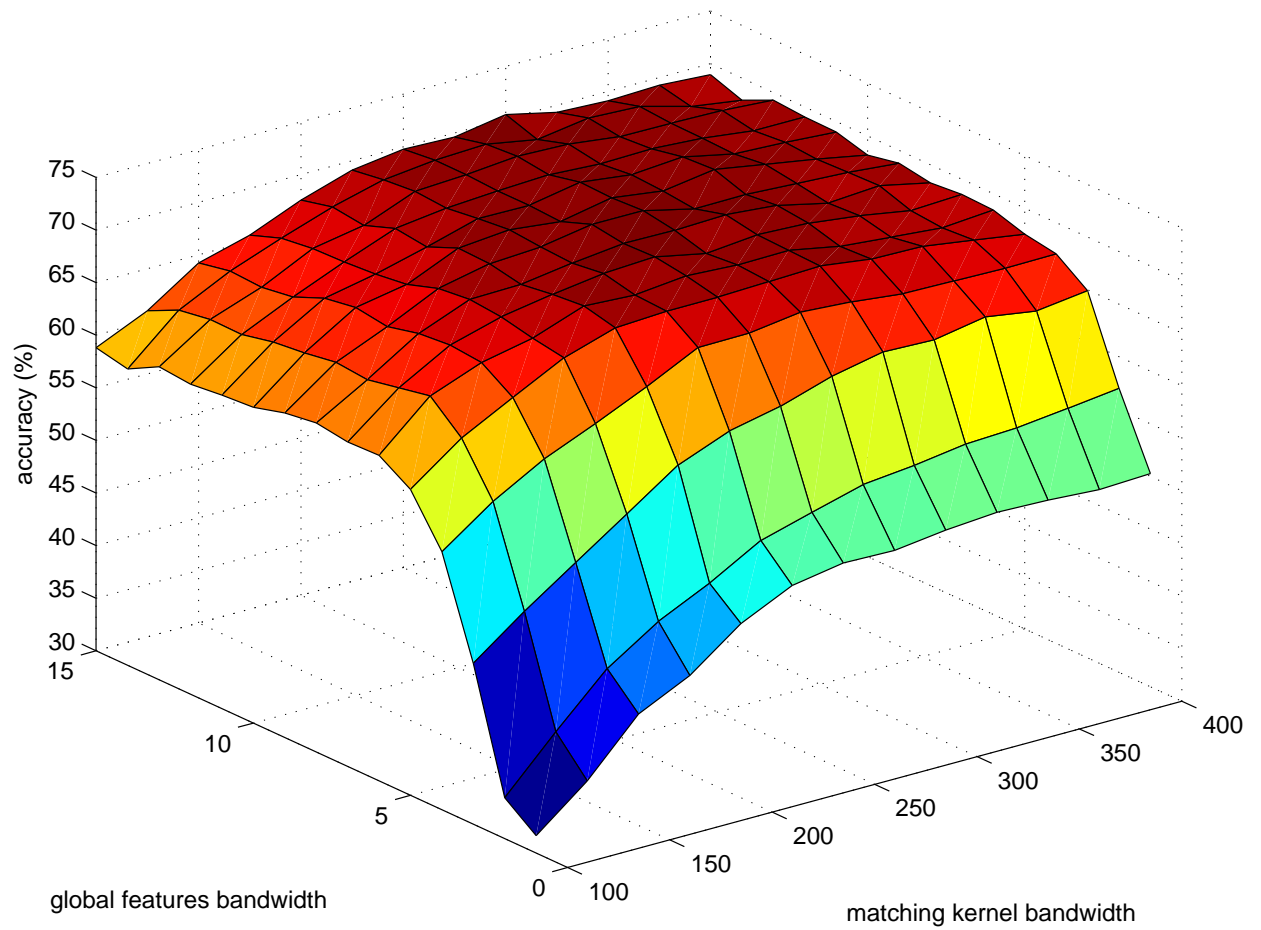


Figure 5.9. Classification accuracies for a weighted polynomial combination of the kernels for global and local features.

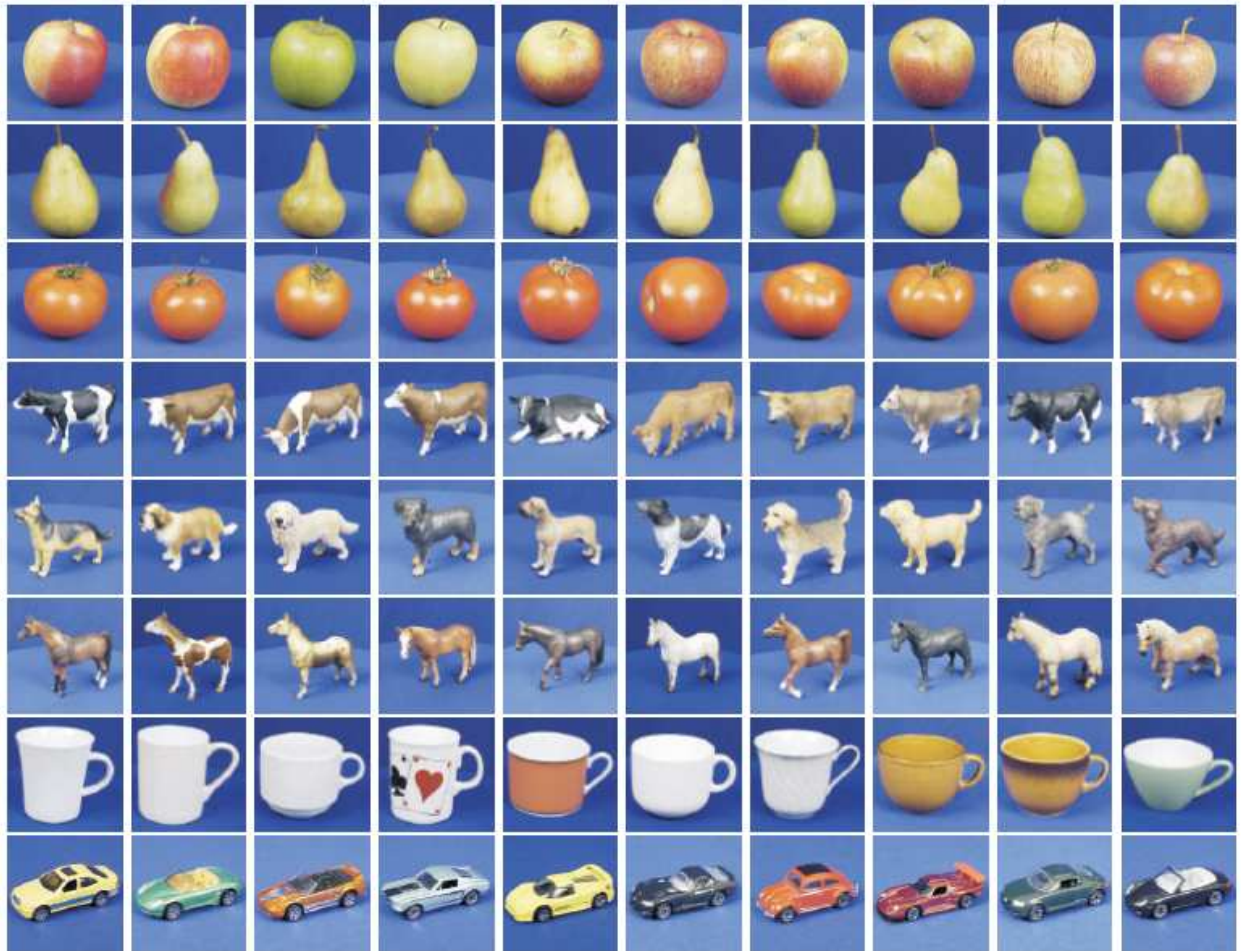


Figure 5.10. Example images from the ETH-80 data set.

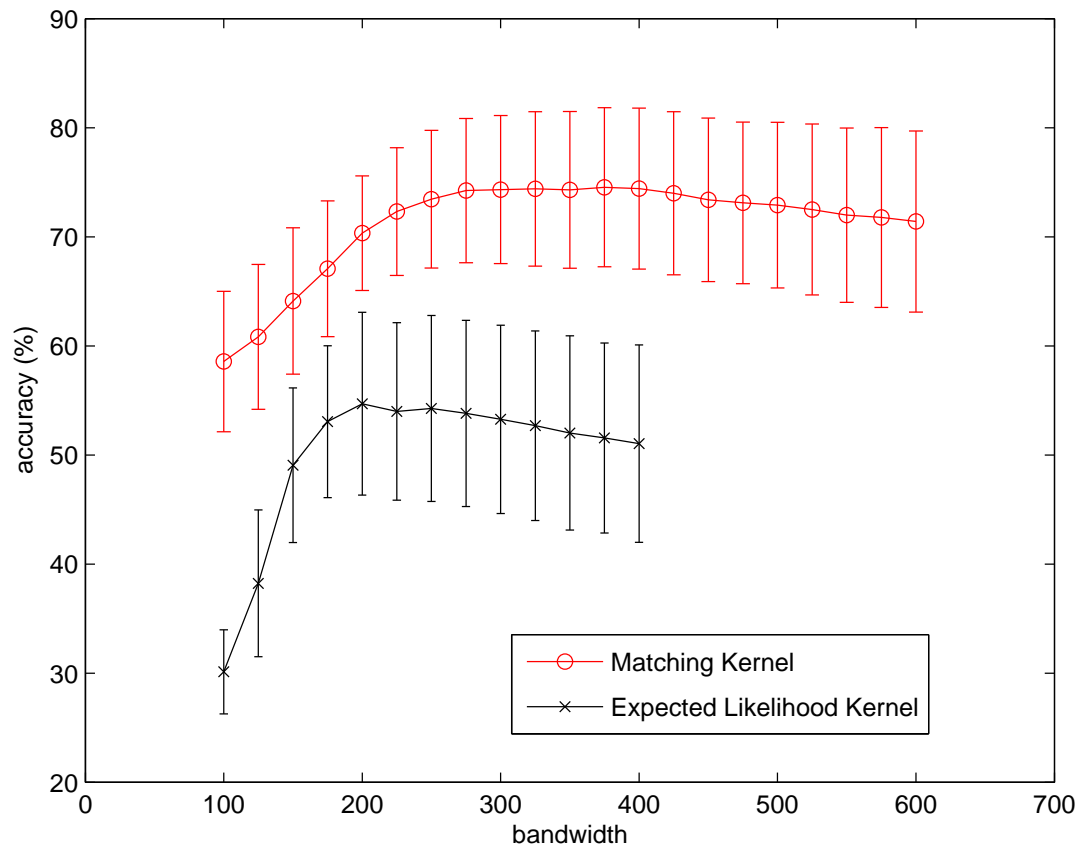


Figure 5.11. Bandwidth, σ , is plotted vs. accuracy on the ETH-80 data set. Results are shown for the matching kernel, and for the expected likelihood kernel.

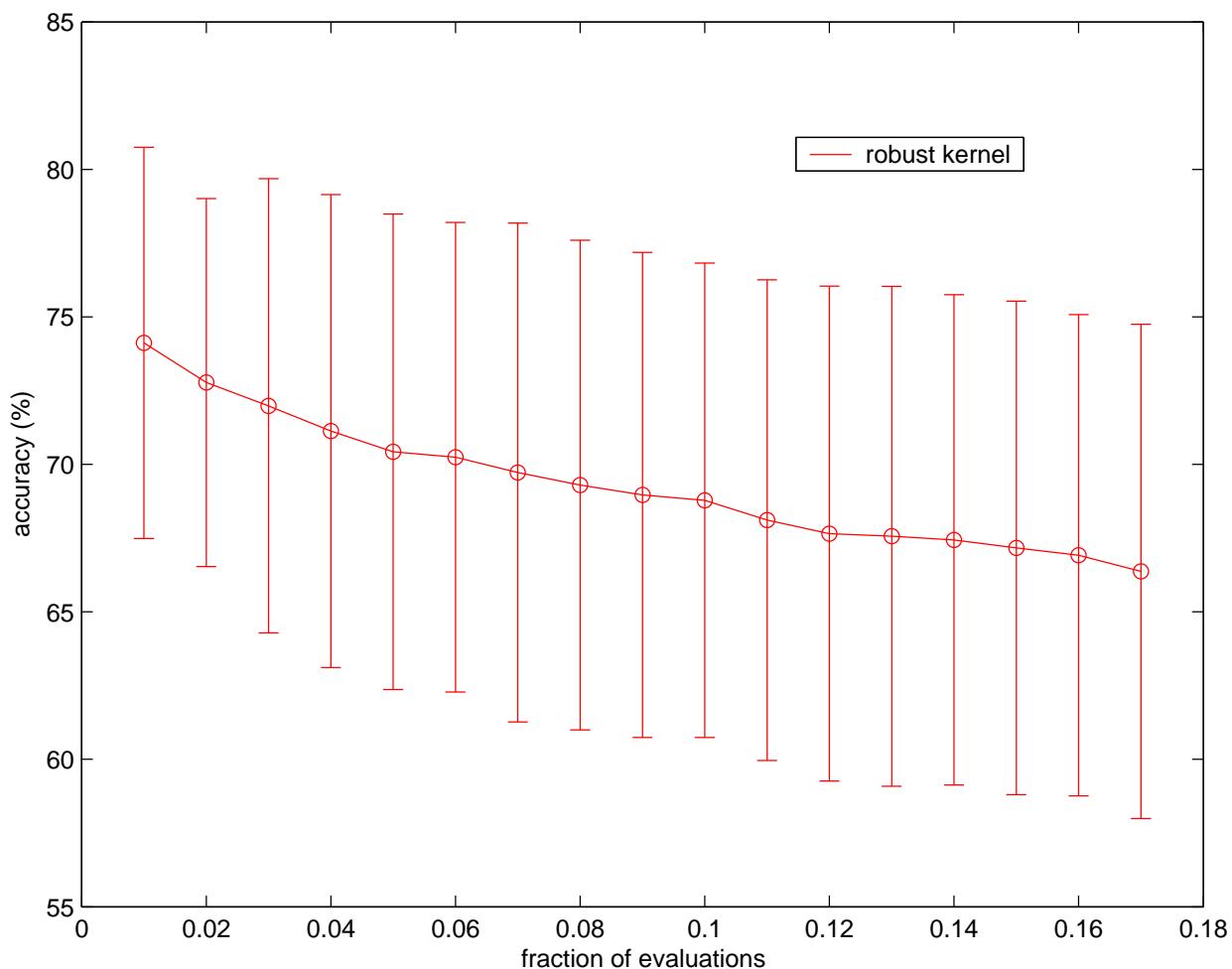


Figure 5.12. Fraction of minor kernel evaluations, β , is plotted vs. accuracy on the ETH-80 data set. The kernel is computed as in equation 5.1.

averaging over a small fraction of the data. The performance seems to monotonically decrease with the number of minor kernel evaluations. This underscores the data dependent nature of the performance of the robust kernel and raises the question of whether interest point detection algorithms might be designed to select sets image patches that would benefit from the robust kernel approach.

CHAPTER 6

CONCLUSION

Interestingly, the best performance using only bags of features of 56.6% on the VPR data set is not so different from that of a related technique in which a maximum likelihood classifier is employed using non-parametric density estimation [28]. In this technique a single distribution over local features is estimated for an *entire class* by estimating a density using all features in the training images. A query image is set to the maximum likelihood estimate computed with the assumption of independence between all features in the image

$$\log p(I|C_i) = \frac{1}{N} \sum_{j=1}^N \log p(x_j|C_i) \quad (6.1)$$

where C_i is a class label. The maximum likelihood technique achieved an accuracy of 52.1% on the VPR dataset.

It is interesting to consider when either of the techniques might perform better than the other. The maximum likelihood classifier is based on an assumption of independence between features in an individual image, and on an assumption that it is appropriate to estimate a distribution over an entire class. This latter assumption may break down in the event that a class consists of a mixture of two or more sub-classes that are drawn from different distributions. In this case, the technique conflates the two distributions. The kernel approach, on the other hand, is based on pairwise comparisons. Because the matching kernel, the expected likelihood kernel, and the robust kernel use a Gaussian RBF kernel as the minor kernel, they in fact have

infinite VC dimension¹. Consequently, the kernel approach can learn more flexible decision boundaries in the presence of sub-classes and will not conflate distributions. A weakness of the kernel approach, however, is closely tied to its strength. By only looking at the vectors present in two images at any time, the approach is limited by the number of features in an image. Eichhorn and Chapelle report that classification performance using local feature kernels increased with the number of features per image [14]. This is not surprising as too few samples will yield a poor estimate of the distribution for a given image.

Additionally, the kernel that combines the matching kernel with global features achieves a maximum performance of approximately 72%, which is slightly better than the best accuracy given in [28], in which the maximum likelihood classifier (equation 6.1) is combined with a SVM classifier over global features using stacking. This approach, therefore, has some of the same expressive capability as meta-learning techniques, despite the significant increase in computational efficiency resulting from the support vector framework.

6.1 Future Work

That the matching kernel outperformed the expected likelihood kernel indicates that a principled approach using non-parametric density estimation is not ideal given the distribution of features in real data. Although using an order statistic increases performance, better modeling of the data themselves may result in more accurate distributions over which the expected likelihood kernel is optimal. Explicitly accounting for subclasses in the data, applying topic models [3, 20], and modeling spurious features in the image are promising places to start.

¹The Vapnik-Chervonenkis dimension is a measure of the capacity of a set of functions to separate sets of data into two different classes [43].

Modeling the data prior to application of the kernel makes use of information regarding the co-occurrence of features in the data, information that was not available when designing a kernel independent of empirical observation. As we make use of additional information sources, feature locations are likely to improve results significantly. Part-based models based on spatial clusters of features is one recently proposed approach [4].

As new sources of information are accounted for, the factorization of the estimated distribution may or may not be structured in such a way that probability product kernels can be computed in closed form. This efficiency is key if the computational advantages of SVMs are to pay off. Consequently, approximations of optimal models must be explored that can be computed efficiently.

BIBLIOGRAPHY

- [1] Barla, A., Franceschi, E., Odone, F., and Verri, A. Image kernels. In *Proceedings of the International Workshop on Pattern Recognition with Support Vector Machines* (2002).
- [2] Bhattacharyya, A. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Math Society* (1943).
- [3] Blei, D., Griffiths, T., Jordan, M., and Tenenbaum, J. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. MIT Press, Cambridge, MA, 2004.
- [4] Bouchard, G., and Triggs, B. Hierarchical part-based visual object categorization. In *IEEE International Conference on Computer Vision and Pattern Recognition* (2005).
- [5] Burges, C. J. C. Geometry and invariance in kernel based methods. In *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, Eds. MIT Press, Cambridge, MA, 1998.
- [6] Burges, C. J. C. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2, 2 (1998), 121–167.
- [7] Chang, C.-C., and Lin, C.-J. LIBSVM: a library for support vector machines. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, 2001.
- [8] Davis, C. S., Gallager, S. M., Berman, M. S., Haury, L. R., and Strickler, J. R. The video plankton recorder (VPR): design and initial results. *Archiv für Hydrobiologie Beiheft Ergebnisse der Limnologie* 36 (1992), 67–81.
- [9] Dietterich, T. G. Ensemble methods in machine learning. In *First International Workshop on Multiple Classifier Systems* (2000), pp. 1–15.
- [10] Dietterich, T. G., and Bakiri, G. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research* 2 (1995), 263–286.
- [11] Dietterich, T. G., Lathrop, R. H., and Lozano-Perez, T. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence* 89, 1–2 (1997), 31–71.

- [12] Dorkó, G., and Schmid, C. Object class recognition using discriminative local features. Submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.
- [13] Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification*, second ed. John Wiley & Sons, Inc, 2001.
- [14] Eichhorn, J., and Chapelle, O. Object categorization with SVM: kernels for local features. Tech. rep., Max Planck Institute for Biological Cybernetics, Tübingen, Germany, 2004.
- [15] Gärtner, T., Flach, P. A., Kowalczyk, A., and Smola, A. J. Multi-instance kernels. In *Nineteenth International Conference on Machine Learning (2002)*, pp. 179–186.
- [16] Goldberger, J., Roweis, S., Hinton, G., and Salakhutdinov, R. Neighborhood components analysis. In *Advances in Neural Information Processing Systems 17*, Lawrence K. Saul, Yair Weiss, and Léon Bottou, Eds. MIT Press, Cambridge, MA, 2005.
- [17] Grauman, K., and Darrell, T. Pyramid match kernels: Discriminative classification with sets of image features. Tech. rep., Massachusetts Institute of Technology - Computer Science and Artificial Intelligence Laboratory, 2005.
- [18] Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley & Sons, Inc, 1986.
- [19] Hettmansperger, T. P., and McKean, J. W. *Robust Nonparametric Statistical Methods*. John Wiley & Sons, Inc, 1998.
- [20] Hofmann, T. Probabilistic latent semantic analysis. In *Uncertainty in Artificial Intelligence (1999)*.
- [21] Jebara, T., and Kondor, R. Bhattacharyya and expected likelihood kernels. In *Conference on Learning Theory (2003)*.
- [22] Jebara, T., Kondor, R., and Howard, A. Probability product kernels. *Journal of Machine Learning Research* 5 (2004), 819–844.
- [23] Kadir, T., and Brady, M. Saliency, scale and image description. *International Journal of Computer Vision* (2001).
- [24] Ke, Y., and Sukthankar, R. PCA-SIFT: A more distinctive representation for local image descriptors. In *IEEE International Conference on Computer Vision and Pattern Recognition (2004)*.
- [25] Kondor, R., and Jebara, T. A kernel between sets of vectors. In *International Conference on Machine Learning (2003)*.

- [26] Kreßel, U. Pairwise classification and support vector machines. In *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, Eds. MIT Press, Cambridge, MA, 1999.
- [27] Leibe, B., and Schiele, B. Analyzing appearance and contour based methods for object categorization. In *IEEE International Conference on Computer Vision and Pattern Recognition* (2003).
- [28] Lisin, D. A., Mattar, M. A., Blaschko, M. B., Benfield, M. C., and Learned-Miller, E. G. Combining local and global image features for object class recognition. In *IEEE Workshop on Learning in Computer Vision and Pattern Recognition* (2005).
- [29] Lowe, D. G. Object recognition from local scale-invariant features. In *Proc. International Conference on Computer Vision* (1999).
- [30] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* (2004).
- [31] Matas, J., Chum, O., Urban, M., and Pajdla, T. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference* (2002).
- [32] Meinicke, P., Twellmann, T., and Ritter, H. Discriminative densities from maximum contrast estimation. In *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. MIT Press, Cambridge, MA, 2003.
- [33] Mikolajczyk, K., and Schmid, C. Indexing based on scale invariant interest points. In *Proc. International Conference on Computer Vision* (2001), pp. 525–531.
- [34] Mikolajczyk, K., and Schmid, C. An affine invariant interest point detector. In *European Conference on Computer Vision* (2002), pp. 128 – 142.
- [35] Mikolajczyk, K., and Schmid, C. A performance evaluation of local descriptors. *Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence* (2004).
- [36] Muslea, I. *Active Learning With Multiple Views*. PhD thesis, University of Southern California, 2002.
- [37] Ravela, S. *On Multi-Scale Differential Features and their Representations for Image Retrieval and Recognition*. PhD thesis, University of Massachusetts Amherst, 2002.
- [38] Rifkin, R., and Klautau, A. In defense of one-vs-all classification. *Journal of Machine Learning Research* 5 (2004), 101–141.

- [39] Schölkopf, B., and Smola, A. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002.
- [40] Schölkopf, B., Smola, A., and Müller, K.-R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* 10 (1998), 1299–1319.
- [41] Seewald, A. K. *Towards Understanding Stacking - Studies of a General Ensemble Learning Scheme*. PhD thesis, Austrian Research Institute for Artificial Intelligence (FAI), 2003.
- [42] Silverman, B. W. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC, 1986.
- [43] Vapnik, V. N. *Statistical Learning Theory*. John Wiley & Sons, Inc, 1998.
- [44] Vidal-Naquet, M., and Ullman, S. Object recognition with informative features and linear classification. In *International Conference on Computer Vision* (2003).
- [45] Wallraven, C., Caputo, B., and Graf, A. B. A. Recognition with local features: the kernel recipe. In *International Conference on Computer Vision* (2003).
- [46] Wolf, L., and Shashua, A. Learning over sets using kernel principal angles. *Journal of Machine Learning Research* 4 (2003), 913–931.