June, 2003

Practical Non-parametric Density Estimation on a Transformation Group for Vision

Erik G Learned-Miller, *University of Massachusetts - Amherst*
Christophe Chefd'hotel

Practical Non-parametric Density Estimation on a Transformation Group for Vision

Erik G. Miller

Computer Science Division
UC Berkeley
Berkeley, California 94720
USA

Christophe Chefd'hotel*

Odyssée Lab
INRIA
06902 Sophia-Antipolis
France

Abstract

It is now common practice in machine vision to define the variability in an object's appearance in a factored manner, as a combination of shape and texture transformations. In this context, we present a *simple* and *practical* method for estimating non-parametric probability densities over a group of linear shape deformations. Samples drawn from such a distribution do not lie in a Euclidean space, and standard kernel density estimates may perform poorly. While variable kernel estimators may mitigate this problem to some extent, the geometry of the underlying configuration space ultimately demands a kernel which accommodates its group structure. In this perspective, we propose a suitable *invariant* estimator on the linear group of non-singular matrices with positive determinant. We illustrate this approach by modeling image transformations in digit recognition problems, and present results showing the superiority of our estimator to comparable Euclidean estimators in this domain.

1. Introduction

It is now common practice in machine vision to model appearance variability in a factored manner as a combination of shape and texture variability [22, 4, 14]. A wide variety of shape models have been proposed, ranging from rigid transformations to arbitrary diffeomorphisms. For many applications, linear models of deformation[1] have provided a good trade-off between flexibility and tractability (both computational and statistical). They also represent an excellent approximation to true perspective projection in a wide range of realistic vision scenarios. Finally, they can be used in combination in a local manner when greater flexibility is desired.

Early applications of linear (affine) models often adopted a *linear invariance* (affine invariance) principle [19], in which two images were considered equivalent if one image could be linearly (affinely) transformed into the other. Recently, it has been more common to assign a cost to such transformations, with a higher cost assigned to "larger" transformations. This has led to a large literature on how to assign the appropriate cost of such transformations, and how to derive a useful notion of distance between two images that satisfies some invariance properties [6, 7, 16].

From a statistical perspective, a natural approach to modeling the distortions or shape change in images is to define a probability density over shape changes for a particular object or set of objects. In this paper, we present a simple, computationally tractable density estimator for linear image transformations. While our estimator is not the first proposed for such sets of transformations (see, e.g., [10, 14]), it offers both the advantage of respecting the underlying *group structure* of the data (to be made precise below), and a simplicity that makes it of practical interest.

1.1. Estimation and transformation groups

It is not uncommon in engineering and machine learning problems for data to have a natural group structure. Perhaps the best known example is in the independent components analysis (ICA) problem, where the transformation that mixes a set of sound sources is unknown and is modeled as an element of the general linear group $\mathrm{GL}(n, \mathbb{R})$, the set of real non-singular $n \times n$ matrices. Gradient based searching in this space of matrices can be done more efficiently by taking advantage of the group structure [1]. This method of attacking the ICA problem has the additional appeal that the algorithm exhibits *uniform performance*, i.e. the solution does not depend upon the mixing matrix [3].

Grenander, who gives a very general approach to probability theory on groups in [5], has recently proposed taking into account the group structure in problems of parameter

*The second author is now a Member of Technical Staff at Siemens Corporate Research, Princeton, New Jersey, USA.

[1] By augmenting linear deformation models with arbitrary translations, we obtain *affine* models.

estimation [8]. A few authors have also looked at *density* estimation on group structures [12, 10] (in particular the group $SO(n)$ of $n \times n$ rotation matrices), and studied the convergence of Fourier series density estimators from a theoretical perspective. Our approach differs from the previous ones in that we focus here on an easily computable and hence practical estimator.

In this paper, we introduce the notion of an *invariant kernel*, and we use such a kernel to produce a probability density from a set of examples, which we refer to as an *invariant kernel density estimate*. The invariance is defined with respect to group structure which holds for the data. We apply our new estimator to the problem of estimating a probability density over image transformations in the context of a factored image model. That is, given a set of image transforms drawn from a fixed but unknown distribution, we wish to estimate a density over the transforms.

We proceed as follows. In Section 2, we review kernel density estimation. In Section 3, we define the notion of invariant kernels, and suggest reasons one might want to take into account the natural group structure of the set of linear shape deformations. In Section 4, we introduce our invariant kernel for the general linear group, and in Section 5 discuss properties of the *invariant density estimator* based upon this kernel. In Section 6, we present preliminary experimental results comparing our estimator to a traditional Gaussian kernel estimator.

2. Kernel Density Estimation

So-called kernel probability density estimators (KDE's), of the form

$$\hat{f}(x; x_1, x_2, ..., x_N) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{h} K\left(\frac{x - x_i}{h}\right),$$

use a set of examples $\{x_1, x_2, ..., x_N\}$ drawn from a random variable X (possibly multi-dimensional), a kernel function K and a bandwidth parameter h, to estimate a probability distribution for X. These estimators play a central role in statistics and machine learning. Perhaps most importantly, such estimators allow the modeling of the complex distributions arising from natural data sources such as images and sounds with a relatively small computational burden.

Rosenblatt [21] and Parzen [17] described such estimators and showed general conditions under which they would converge to the true distribution as the sample size grows and the kernel bandwidth shrinks. In this paper, our goal is to model distributions over linear image transformations, which can be conveniently represented as two by two matrices with positive determinant[2]. It is tempting to use a kernel density estimator "out of the box" to estimate a density over transformation matrices by treating each matrix as an element of a four-dimensional vector space. In the limit of an infinite number of samples, this estimator will converge to the true probability density.

However, the asymptotic convergence of a density estimator does not imply it will work well for practical density estimation. In particular, since the distribution of matrices tends to be more concentrated (in a Euclidean sense) for matrices with determinant less than one, and less concentrated for matrices with determinant greater than one, one might try a variable kernel estimator (as described in [13, 2]) to improve the rate of convergence. While such adaptive density estimates may converge more rapidly to the true distribution in many cases, they have more parameters and thus may have relatively high variance. This is particularly relevant when we have a limited amount of data. We propose an alternative: to develop better non-parametric estimators in low data scenarios by taking advantage of the intrinsic properties of the set of transformation matrices.

3. Group structure and invariance

Traditionally in kernel density estimators, kernels are functions of the difference between the coordinates of two points x and x_i in a Euclidean space. As a result, the kernels are invariant to translation. However, not all probability densities are well modeled by such "Euclidean" KDE's. In particular, if there is a specific group structure and geometry in a set of data, for example if samples have been drawn from a set of transformation matrices, then other types of estimators may be more appropriate.

3.1. Invariant kernels

Consider a more general kernel function

$$K(t; a) = f(D(t, a)),$$

where a is a parameter that defines the "center" of the kernel and $K(t; a)$ takes on a value as a function of how different t is from a. More formally, some function $D(t, a)$ determines the difference (not necessarily a vector difference) between t and a, and the kernel K is some function of this difference.

For a group G, we consider a *group difference* function defined by

$$D_G(t, a) = t^{-1} \circ a,$$

where $\circ$ is the group operator, and t^{-1} is the group inverse of t. The group difference function is invariant to the application of a fixed group element to both arguments. That

[2] A straight-forward extension of the ideas in this paper allow the modeling of "reflecting" linear transformations, i.e. two by two matrices with negative determinants.

is,

$$\begin{aligned} D_G(b\circ t, b\circ a) &= [b\circ t]^{-1}\circ b\circ a \\ &= t^{-1}\circ b^{-1}\circ b\circ a \\ &= t^{-1}\circ a \\ &= D_G(t,a). \end{aligned}$$

Predictably, we define a (left) *invariant kernel function*, with respect to a group G, as one that satisfies

$$K(t;a) = K(b\circ t; b\circ a).$$

Any kernel which is a function of the group difference will satisfy this property automatically, since it is satisfied by the group difference. For example, the common unit variance one-dimensional Gaussian kernel

$$K(t;a) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(t-a)^2}$$

is invariant with respect to the group $G = \mathbb{R}$ when the group operator $\circ$ is "+", since

$$\begin{aligned} K(b+t; b+a) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(b+t-(b+a))^2} \\ &= \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(t-a)^2} \\ &= K(t;a). \end{aligned}$$

Here, the group difference is simply vector (or in this case, scalar) subtraction. Since the kernel is a function of this difference, it is invariant with respect to the group of translations in one dimension.

Certain applications, however, may suggest a notion of difference other than the vector difference which characterizes Euclidean spaces [5]. In particular, when points in a space are combined not through addition, but with some other operator, it is frequently advantageous to use this operator in the definition of the difference between points. If we consider the set of real 2×2 non-singular matrices with positive determinant (denoted $\mathrm{GL}^+(2,\mathbb{R})$ in the following) which can be used to rotate, shear, or scale images, we notice that its elements are naturally combined via matrix multiplication. But the question remains, what property is it that demands that we treat these matrices as elements of $\mathrm{GL}^+(2,\mathbb{R})$ rather than 4-vectors in a Euclidean space? To answer this question, we must look at the specific application of our estimator, the analysis of handwritten digits.

3.2. A Random Transform Process

When people write digits and letters, there is natural variability in the pose of each letter. The pose varies both within and across writers. One can think of the pose of each letter as a transformation away from some canonical pose for that letter. Thus, one may consider the writing of a set of handwritten 2's as a random process, that among other things, produces samples of a random transform variable.

Consider for a moment the image of a two shown as "Image A" in Figure 1. We may choose to represent this image as the transformation of some other image in a canonical form, such as Model 1 of Figure 1. Then, the representation of Image A could take the form of a pair, $(digit, transform) = (2, \mathbf{T}_A^1)$, where the digit identifies the base type of the image (a two), and $\mathbf{T}_A^1$ is the 2x2 matrix (with positive determinant) which, when applied to Model 1 will produce Image A. However, we could just as well choose Model 2 as the canonical form of the character "2", in which case the representation of Image A would be $(digit, transform) = (2, \mathbf{T}_A^2)$, indicating that a different transform, which depends upon the Model 2, is needed to produce Image A. The key point here is that the transformation is not an inherent part of a single image, but is defined only relative to the digit model.

Now suppose we want to define a difference between a transform associated with Image A and a transform associated with Image B. We make the following demand of our difference operator for transformations: that the difference between the transformations for two characters be *invariant* to the choice of model. More generally, let $\mathbf{S}$ represent the transformation from Model 1 to Model 2. Our requirement on the difference operator can be written as

$$\begin{aligned} D(\mathbf{T}_A^1, \mathbf{T}_B^1) &= D(\mathbf{T}_A^2, \mathbf{T}_B^2) \\ &= D(\mathbf{S}\cdot\mathbf{T}_A^1, \mathbf{S}\cdot\mathbf{T}_B^1), \end{aligned} \tag{1}$$

where "$\cdot$" denotes the matrix multiplication (it will be omitted in the following). It is easy to see that adopting matrix multiplication as the group operator, and the (non-negative determinant) matrices as the group elements, satisfies this demand. Thus, the invariance of the difference operator naturally suggests the group structure for this type of data.

We point out that choosing a model with which to define relative transforms is tantamount to choosing an "origin", i.e. an identity element, for the group of transformations. Then Eq. 1 can be viewed as invariance to the choice of origin for the set of transformation coordinates. Thus, any function of the group difference operator is in this sense *coordinate free* or *intrinsic*.

Finally, a simple numerical example may clarify the intuitive desire for a non-Euclidean difference. The matrices

$$\begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \text{and} \begin{bmatrix} 7.07 & 7.07 \\ -7.07 & 7.07 \end{bmatrix}$$

have a Euclidean difference with much greater magnitude than the matrices

$$\begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \text{and} \begin{bmatrix} 0.0707 & 0.0707 \\ -0.0707 & 0.0707 \end{bmatrix},$$

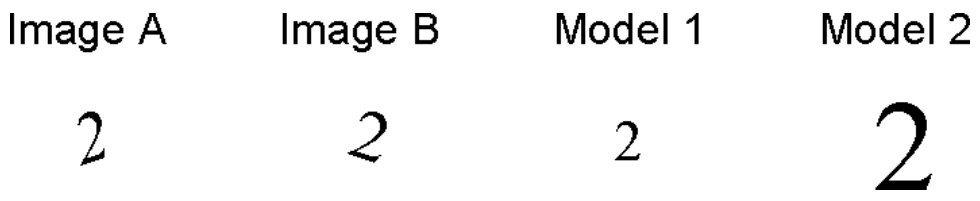

Figure 1: The transforms associated with an image of a character (Image A or Image B) is a function not only of the image itself, but of the model to which it is compared (Model 1 or Model 2). By requiring that the difference between two image transforms be invariant to the choice of model, we naturally impose a specific group structure on the set of transformations.

despite the fact that each pair of matrices "differs" only by a rotation of 45 degrees. The linear group difference function does not suffer from this property.

Since the linear group difference function for non-singular 2x2 matrices is invariant to the application of a group operation any kernel based upon this difference will also be invariant. We now discuss such a kernel.

4. Invariant kernels for $\mathrm{GL}^+(2,\mathbb{R})$

We propose the following invariant kernel function for $\mathrm{GL}^+(2,\mathbb{R})$:

$$K(\mathbf{T};\mathbf{A}) = \frac{1}{C(h)} e^{-\frac{1}{2h}\|\log(\mathbf{T}^{-1}\mathbf{A})\|_F^2},$$

where h is a kernel bandwidth parameter, C is a normalization constant that depends upon the bandwidth, log is a *matrix logarithm*, and $\|\cdot\|_F$ is the Frobenius norm, the square root of the sum of the products of the matrix components by their complex conjugates. This kernel is a function of the natural group difference $\mathbf{T}^{-1}\mathbf{A}$ between two matrices $\mathbf{T}$ and $\mathbf{A}$. Improving on the kernel discussed in [15], it is also symmetric. That is, $K(\mathbf{T};\mathbf{A}) = K(\mathbf{A};\mathbf{T})$. Furthermore, combining invariance and symmetry, we have $K(\mathbf{T};\mathbf{I}) = K(\mathbf{T}^{-1};\mathbf{I})$, where $\mathbf{I}$ is the identity matrix. A discussion of the matrix logarithm and a derivation of these properties are found in the appendix.

By construction, the kernel function is invariant since:

$$\begin{aligned} K(\mathbf{BT};\mathbf{BA}) &= \frac{1}{C} e^{-\frac{1}{2h}\|\log((\mathbf{BT})^{-1}\mathbf{BA})\|_F^2} \\ &= \frac{1}{C} e^{-\frac{1}{2h}\|\log(\mathbf{T}^{-1}\mathbf{B}^{-1}\mathbf{BA})\|_F^2} \\ &= \frac{1}{C} e^{-\frac{1}{2h}\|\log(\mathbf{T}^{-1}\mathbf{A})\|_F^2} \\ &= K(\mathbf{T};\mathbf{A}). \end{aligned}$$

Notice that the definition of such a kernel is not restricted to two by two matrices, and could be applied to $\mathrm{GL}^+(n,\mathbb{R})$ for arbitrary n.

However, it is not enough that our kernel produce the same value for a point $\mathbf{T}$ relative to the kernel parameter $\mathbf{A}$ under transformation of these values by $\mathbf{B}$. Since our goal is to use the kernel as *a probability law* to describe the probability of events, we must also have, for an event E, that

$$\begin{aligned} Prob(E) &= \int_{\mathbf{T}\in E} \frac{1}{C} e^{-\frac{1}{2h}\|\log(\mathbf{T}^{-1}\mathbf{A})\|_F^2} d\mu \\ &= \int_{\mathbf{T}\in \mathbf{B}\cdot E} \frac{1}{C} e^{-\frac{1}{2h}\|\log(\mathbf{T}^{-1}\mathbf{BA})\|_F^2} d\mu \\ &= Prob(\mathbf{B}\cdot E), \end{aligned}$$

under some measure μ.

To achieve this, we must define the kernel function as a density not relative to the measure obtained from the standard volume element $d\mathbf{T} = \prod_{1\le i,j\le 2} d\mathbf{T}_{i,j}$, but relative to an invariant measure on $\mathrm{GL}^+(2,\mathbb{R})$.

An invariant measure exists on any locally compact group, such that $\mathrm{GL}^+(n,\mathbb{R})$, and is called the Haar measure [18]. In our case, it is directly derived from a (left) invariant volume element. Its expression in terms of $d\mathbf{T}$ is given by

$$\frac{1}{|\mathbf{T}|^n} d\mathbf{T},$$

where $|\cdot|$ denotes the determinant of $\mathbf{T}$. We refer the reader to [18] for a complete discussion of invariant measures on groups.

Such a condition requires that

$$\int_{\mathbf{T}\in \mathrm{GL}^+(2,\mathbb{R})} \frac{1}{C} e^{-\frac{1}{2h}\|\log(\mathbf{T}^{-1}\mathbf{A})\|_F^2} \frac{1}{|\mathbf{T}|^2} d\mathbf{T} = 1. \qquad (2)$$

At this point, we make a significant assumption, which is that our kernel is integrable under the given invariant measure. Assuming this property holds (which did not prove[3]), we must have that

$$C = \int_{\mathbf{T}\in \mathrm{GL}^+(2,\mathbb{R})} e^{-\frac{1}{2h}\|\log(\mathbf{T}^{-1}\mathbf{A})\|_F^2} \frac{1}{|\mathbf{T}|^2} d\mathbf{T}.$$

Thus for a fixed bandwidth h, C is a constant, and is not dependent upon where the kernel is centered. Finally, note

[3] But seems confirmed numerically by Monte-Carlo integration experiments.

that while we define the invariance properties of our kernel with respect to the invariant measure on $\mathrm{GL}^+(2,\mathbb{R})$, we can give the kernel density with respect to $d\mathbf{T}$ as well, which can be read directly from Eq. 2:

$$K_E(\mathbf{T};\mathbf{A}) = \frac{1}{C} e^{-\frac{1}{2h}\|\log(\mathbf{T}^{-1}\mathbf{A})\|_F^2} \frac{1}{|\mathbf{T}|^2}. \quad (3)$$

5. A density estimator for $\mathrm{GL}^+(2,\mathbb{R})$

Suppose now that we have a set of samples $\mathbf{T}_1, \mathbf{T}_2, ..., \mathbf{T}_N$ from a matrix random variable defined over $\mathrm{GL}^+(2,\mathbb{R})$. Armed with our invariant kernel, we can form a density estimate using the set of samples in a style similar to the Parzen estimate:

$$f(\mathbf{U};\mathbf{T}_1,\mathbf{T}_2,...,\mathbf{T}_N) = \frac{1}{N}\sum_{i=1}^{N} K(\mathbf{U};\mathbf{T}_i).$$

We also refer to this density estimator as invariant, since

$$\begin{aligned} f(\mathbf{BU};\mathbf{BT}_1,\mathbf{BT}_2,...,\mathbf{BT}_N) &= \frac{1}{N}\sum_{i=1}^{N} K(\mathbf{BU};\mathbf{BT}_i) \\ &= \frac{1}{N}\sum_{i=1}^{N} K(\mathbf{U};\mathbf{T}_i) \\ &= f(\mathbf{U};\mathbf{T}_1,\mathbf{T}_2,...,\mathbf{T}_N). \end{aligned}$$

That is, premultiplication of the samples $\mathbf{T}_i$ used for estimation and the test point $\mathbf{U}$ being evaluated by any matrix in $\mathrm{GL}^+(2,\mathbb{R})$ does not change the probability density assigned to the point. Note again that this argument extends immediately to *n* by *n* matrices, making these results applicable to general linear groups in arbitrary dimension.

This estimator looks common enough until one considers the extent of the kernels in a Euclidean space. Kernels centered around $\mathbf{T}_i$ with small determinant are "peaky" and have low variance (according to a Euclidean measure). At the same time, kernels for $\mathbf{T}_i$ with large determinant are spread out and have relatively high variance (again in a Euclidean sense). Finally, these kernels have very different shape in the sense that they are not translations or scalings of each other in Euclidean space. However, they do have the proper invariance properties. What remains is to see whether they perform well in practice.

6. Comparing density estimates

We compared our density to a traditional kernel estimator, using fixed spherical Gaussian kernels. We performed three different types of experiments. In the first experiment, we built a hamstrung handwritten digit-classifier using *only* information about the linear transformations necessary to align a test character with each model. In a second experiment, we modified a factored classifier discussed in previous work [15], using the new invariant estimator to build the transformation density. Finally, we did a simple comparison of likelihoods of a hold-out sample under the two types of density estimators. We stress that all of these tests are designed to examine how well the transforms are modeled, rather than to maximize performance of a digit classifier. Before describing the experiments, we discuss the source of the random linear transformations being modeled.

6.1. Factored character models

In [14], we presented a factored model of handwritten digits. A quantity proportional to the posterior density of a digit class given an image (with a uniform class prior) was computed as

$$p(c|I) \sim p(I_L|c) \cdot p(\mathbf{T}|c), \quad (4)$$

where I_L is the "latent image" that results from aligning an image I to a model and $\mathbf{T}$ can be thought of as the transform that produced the observed image I from the latent image. Figure 2 shows a set of handwritten zeroes (observed images) on the left. The result of aligning these images to each other is a set of latent images, shown on the right of the figure.

By aligning a set of images from a single class to each other (we call this *congealing* [14]), we implicitly define a set of transforms (mapping the aligned latent images back to the observed image). We can use these "training transforms" to define a density. To classify a test image, we align the image to each model (which in this case is a set of congealing-aligned images) and maximize the likelihood in (4).

6.2. The transformation-only classifier

Since our goal in this work is not to classify digits, but to test various models of transforms, we modified our classifier to *completely ignore the latent image term* in (4). The classifier thus worked as follows. For each test image, align that image as well as possible to each digit model. For each digit model, this results in some transform $\mathbf{T}_i$. Evaluate the likelihood of this transform under the transform density for each class, and choose the class with the highest likelihood.

To illustrate, consider a test character "6". Suppose we align this test character to the "9" model. The transform which optimizes this alignment is a 180 degree rotation. Under the data derived model of typical transformations for "9"s, such a transformation is very unlikely, and hence, the likelihood that the test character is in fact a nine would be assigned a very small value. While this example is trivial for either a traditional density estimator or the invariant estimator, the hypothesis is that a good transform density esti-

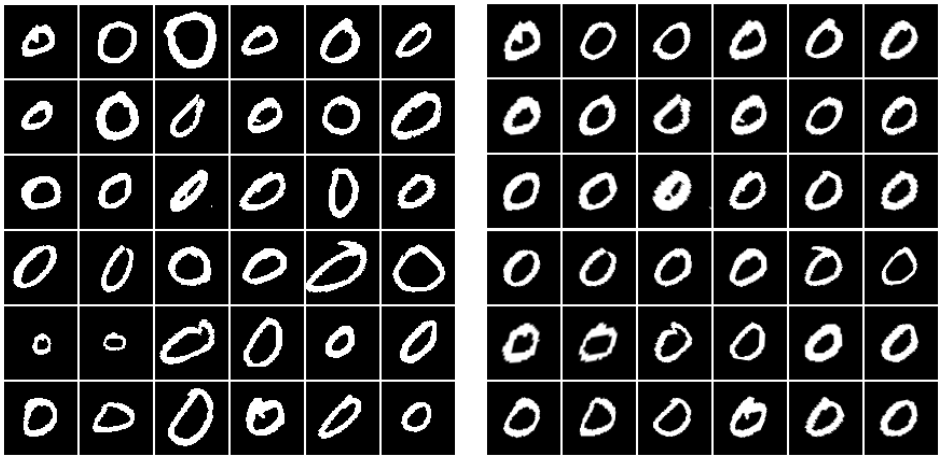

Figure 2: On the left are a set of handwritten zeroes. On the right are the same characters after having been linearly transformed (and shifted) to maximize a criterion of their alignment. Such a "congealing" process produces a set of latent images, and also, implicitly a sample of the random transform process that characterizes the shape variability of the observed images. It is this process of linear shape variability, represented as matrices, that we wish to characterize with a probability density function.

mator will be more successful in making subtler distinctions of this type.

For this experiment we used varying number of transforms to define the model transform density for each digit. We then tested on 100 examples of each digit. The classification accuracy is shown in Figure 3. The dashed line represents the invariant estimator, and the solid line the Parzen estimator with Gaussian kernels. Note that both classifiers did significantly better than random (10%) despite working with only information about the transform variable, but the invariant density consistently outperformed the Gaussian estimator. The bandwidth parameter of both classifiers was chosen to maximize the accuracy for each data set size.

Since the Gaussian kernel estimator eventually converges to the true distribution (assuming it's smooth, etc.), we would expect it to do as well or possibly better than the invariant estimate as the number of training examples goes to infinity, but with 1000 training examples per model density, the invariant estimator still has a clear advantage. We again emphasize that the goal of this experiment is to demonstrate the superiority of the transform density estimator, not to break records for digit recognition.

6.3. Learning from one example

In a second experiment, we applied the new estimator to the problem of "learning from one example" as described in [14]. In these experiments, a factored model of each digit, consisting of a latent image model and a transformation model was again developed. The crude latent image model was taken from a single image of each digit class (hence learning from one example). The model of transforms, however, was developed by congealing (aligning) handwritten *letters* of the same class, and collecting the resulting transforms. The assumption underlying this technique is that spatial variations such as rotations and scalings should have similiar statistics across character classes. The final performance of these classifiers is then substantially dependent upon the quality of the transformation model developed from the set of transforms "borrowed" from the letter classes. The purpose of this experiment was to see if classification of digits could be improved by using the new transformation density to produce a transform model.

In these experiments, the Gaussian kernel density produced an accuracy of 88.2%, while the invariant estimator improved this result to 89.3%. The new estimator also improved upon the previous estimator discussed in [15], that did not take advantage of the properties of the matrix logarithm. This previous estimator achieved an accuracy of 88.6%. While these differences are small, it should be remembered that we are probably approaching the performance limits of a single example classifier, so large gains should not be expected.

6.4. Maximum likelihood

Finally, we evaluated the density estimators by computing the average log likelihood of a test sample under each density. Using 50 "training" examples to define each nonparametric density, we maximized the likelihood of another 50 hold-out samples by optimizing the bandwidth parame-

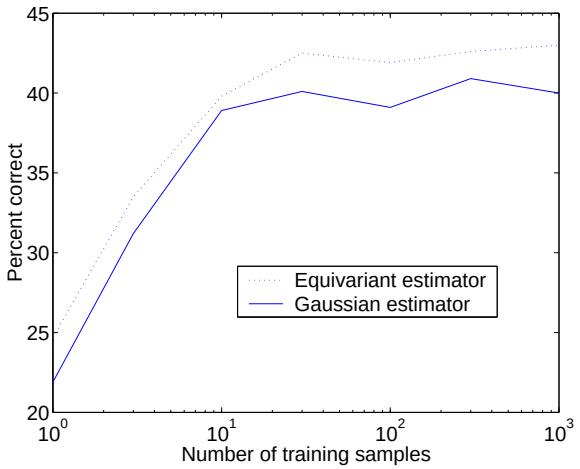

Figure 3: The figure shows the classification accuracy on a digit recognition task using *only* information about the transformation needed to align a test character to each model. The solid line gives the performance using a Parzen estimate with Gaussian kernels, while the dotted line shows the improved performance resulting from the invariant estimator.

ter. We found that for these sample sizes, the likelihoods were consistently higher for the invariant density estimator. Typical values for average log likelihoods were about 0.2 for the Gaussian kernel density and about 1.7 for the invariant estimator. For very small sample sizes ($N = 10$ for training) we found that the Gaussian kernel estimator slightly outperformed the invariant estimator. Since our bandwidth optimizer was much more effective for the Gaussian density, we suspect this phenomenon could have been a result of "overfitting" to the hold-out set[4].

7. Summary

We have presented a probability density estimator that is adapted to the structure of $\mathrm{GL}^+(n,\mathbb{R})$. Since the basic apparatus depends only upon the generic properties of matrix transformation groups, we believe the same ideas could apply easily to larger or smaller sets of simple geometric transformations (affine, rotations, ...) in arbitrary dimension. We applied our estimator to a problem in pattern recognition and showed improved results relative to a traditional Parzen estimator with Gaussian kernels. Note that a rigorous theoretical study of the kernel properties, in particular verifying its integrability, remains to be done.

[4]We shall repeat these experiments with distinct hold-out and test sets as soon as possible.

Appendix: Matrix logarithms

The characteristics of our kernel K follow directly from the properties of the function:

$$d(\mathbf{A};\mathbf{B}) = ||\log(\mathbf{A}^{-1}\mathbf{B})||_F.$$

In a very informal interpretation, the use of a matrix logarithm can be seen as an attempt to "linearize" the structure of the transformation group. We show below that the symmetry of $d(\mathbf{A};\mathbf{B})$, and thus $K(\mathbf{A};\mathbf{B})$, is a direct consequence of this choice.

When it exists, we call logarithm of a matrix $\mathbf{X}$ any solution $\mathbf{A}$ of

$$e^{\mathbf{A}} = \mathbf{X}.$$

Any nonsingular matrix $\mathbf{X}$ (i.e. any matrix in $\mathrm{GL}(n,\mathbb{R})$) has matrix logarithms (in general infinitely many). Note that matrix logarithms are generally complex valued. For a rigorous definition and complete discussion of this matrix function, we refer to [11, 9].

In this work, we consider the *primary matrix logarithm* [11] evaluated using an implementation of the Schur decomposition method described in [9] (the principal branch of the scalar complex logarithm is used). This matrix logarithm satisfies some useful properties:

- $\log(\mathbf{I}) = \mathbf{0}$,
- $\log(\mathbf{A}^T) = (\log(\mathbf{A}))^T$,
- $\log((\overline{\mathbf{A}})^{-1}) = -(\overline{\log(\mathbf{A})})$,
- $\log((\mathbf{A}^{-1})^*) = \log((\mathbf{A}^*)^{-1}) = -(\log(\mathbf{A}))^*$,

where $\overline{\mathbf{A}}, \mathbf{A}^T, \mathbf{A}^* = \overline{\mathbf{A}}^T$ denote respectively the complex conjugate, the matrix transpose and the Hermitian adjoint. However, note that matrix logarithms do not share all the properties of their scalar counterpart (for instance, we will not have $\log(\mathbf{AB}) = \log(\mathbf{A}) + \log(\mathbf{B})$ in general).

Combining the properties of the Frobenius norm, the group difference and the previous results, for all $\mathbf{A},\mathbf{B} \in \mathrm{GL}^+(n,\mathbb{R})$, we can check that $d(\mathbf{A};\mathbf{B})$ satisfies:

- (Positiveness) $d(\mathbf{A};\mathbf{B}) \geq 0$.
- (Invariance) Given $\mathbf{X} \in \mathrm{GL}^+(n,\mathbb{R})$, $d(\mathbf{XA};\mathbf{XB}) = d(\mathbf{A};\mathbf{B})$, since $(\mathbf{XA})^{-1}\mathbf{XB} = \mathbf{A}^{-1}\mathbf{B}$.
- (Symmetry) $d(\mathbf{A};\mathbf{B}) = ||\log(\mathbf{A}^{-1}\mathbf{B})||_F = ||\log((\mathbf{B}^{-1}\mathbf{A})^{-1})||_F = ||-\overline{\log(\mathbf{B}^{-1}\mathbf{A})}||_F = ||\log(\mathbf{B}^{-1}\mathbf{A})||_F = d(\mathbf{B};\mathbf{A})$.

Another interesting point is to consider the one-dimensional case: $\mathrm{GL}^+(1,\mathbb{R}) =]0,+\infty[$. The usual real scalar logarithm can be used, and for $C(h) = \sqrt{2\pi h}$, the Euclidean density $a \mapsto K_E(a;b)$ corresponding to $K(a;b)$ (see

end of Section 4) reduces to the probability density of a *log-normal* distribution:

$$f(a) = \frac{1}{\sigma a\sqrt{2\pi}} e^{-\frac{(\log a - m)^2}{2\sigma^2}},$$

with $m = \log(b)$ and $\sigma^2 = h$.

Acknowledgements

EGM would like to thank Adrian Corduneanu and Tommi Jaakkola for helpful discussions related to this work.

References

[1] Amari, S. Natural gradient works efficiently in learning. *Neural Comp., 10*. pp.251-276. 1998.

[2] Breiman, L., Meisel, W. and Purcell, E. Variable kernel estimates of multivariate densities. *Technometrics, 19*. pp. 135-144. 1977.

[3] Cardoso, J. The invariant approach to source separation. *Proceedings of the International Symposium on Nonlinear Theory and Applications, 1*. pp. 55-60. 1995.

[4] Frey, B. and Jojic, N. Transformed component analysis: Joint estimation of spatial transformations and image components. *In Proceedings of the IEEE International Conference of Computer Vision*. 1999.

[5] Grenander, U. *Probabilities on Algebraic Structures*. Almqvist and Wiksell, Stockholm and John Wiley and Sons, New York. 1963.

[6] Grenander, U. *General Pattern Theory*. Oxford Sciences Publications. 1993.

[7] Grenander, U. and Miller, M. I. Computational anatomy: an emerging discipline. *Quart. Appl. Math., 56*. pp. 617-694. 1998.

[8] Grenander, U. and Miller, M. I. and Srivastava, A. Hilbert-Schmidt lower bounds for estimators on matrix Lie groups for ATR. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*. pp. 790-802. 1998.

[9] Golub, G. H. and Van Loan, C. F. *Matrix Computations*. North Oxford Academic. 2nd edition. 1986.

[10] Hendriks, H. Nonparametric estimation of a density on Riemannian manifold using Fourier expansions. *The Annals of Statistics, 18*. pp. 832-849. 1990.

[11] Horn, R. A. and Johnson, C. R. *Topics in Matrix Analysis*. Cambridge University Press. 1991.

[12] Kim, P. T. Deconvolution density estimation on SO(N). *Annals of Statistics 26*. pp. 1083-1102, 1998.

[13] Loftsgaarden, D. O. and Quesenberry, C. P. A nonparametric estimate of a multivariate density function. *The Annals of Mathematical Statistics, 36*. pp. 1049-1051. 1965.

[14] Miller, E. G., Matsakis, N., Viola, P. A. Learning from one example through shared densities on transforms. *IEEE Conference on Computer Vision and Pattern Recognition*. 2000.

[15] Miller, E. G. Learning from one example in machine vision by sharing probability densities. Ph.D. thesis. MIT. 2002.

[16] Miller, M. I. and Younes L. Group actions, homeomorphisms, and matching: a general framework. *International Journal of Computer Vision, 41*. pp. 61-84. 2001.

[17] Parzen, E. On estimation of a probability density function and mode. *Annals of Mathematical Statistics, 33*. pp. 1065-1076. 1962.

[18] Santaló, L. A. *Integral Geometry and Geometric Probability.* Addison-Wesley, Reading, MA. 1976.

[19] Simard, P., LeCun, Y., and Denker, J. Efficient pattern recognition using a new transformation distance. *In Advances in Neural Information Processing Systems 5*, pp. 51-58. 1993.

[20] Srivastava, A. and Klassen, E. Monte Carlo extrinsic estimators of manifold-valued parameters. *IEEE Transactions on Signal Processing, 50*. pp. 299-308. 2002.

[21] Rosenblatt, M. Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics, 27*. pp. 832-837. 1956.

[22] Vetter, T., Jones, M., and Poggio, T. A bootstrapping algorithm for learning linear models of object classes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 40-46. 1997.