NIPS*97 Submission
Category: Visual Processing
Preference: Oral

Using Mixtures of Factor Analyzers for Segmentation and Pose Estimation

Geoffrey E. Hinton and Michael Revow
Department of Computer Science
University of Toronto
Toronto, Ontario, M5S 1A4, Canada
hinton@cs.toronto.edu, revow@cs.toronto.edu

Abstract

To read a hand-written digit string, it is helpful to segment the image into separate digits. Bottom-up segmentation heuristics often fail when neighboring digits overlap substantially. We describe a system that has a stochastic generative model of each digit class and we show that this is the only knowledge required for segmentation. The system uses Gibbs sampling to construct a perceptual interpretation of a digit string and segmentation arises naturally from the "explaining away" effects that occur during Bayesian inference. By using *conditional* mixtures of factor analyzers, it is possible to extract an explicit, compact representation of the instantiation parameters that describe the pose of each digit. These instantiation parameters can then be used as the inputs to a higher level system that models the relationships between digits. The same technique could be used to model individual digits as redundancies between the instantiation parameters of their parts.

1 Introduction

We use low-resolution overlapping digits to illustrate the advantages of generative models for image segmentation and pose estimation. Some inadequacies of our density models for individual digits are then addressed and a hierarchical model is proposed.

An image of a handwritten digit is a point in a continuous intensity space that has as many dimensions as there are pixels. For digits within the same class, the points lie on or near a smooth, low-dimensional manifold. For smooth intensity images, this manifold is locally linear because small changes in the position, size, orientation or deformation of a digit lead to approximately linear changes in pixel intensities (Simard, LeCun and Denker, 1993). It is therefore sensible to model the manifold using a mixture of locally linear patches. A simple way to do this is to use a mixture of principal component analyzers (PCA). The mixture is fitted to data using the K-means algorithm in an outer loop that assigns each datapoint to the closest PCA subspace, and singular value decomposition in an inner loop that refits the subspace of each PCA analyzer to the datapoints assigned to it.

Unfortunately, PCA is not a proper density model because projecting the data onto the subspace spanned by the principal components is equivalent to assuming that the variances in all directions orthogonal to this subspace are infinitesimal compared with the variances within the subspace. Factor analysis resembles PCA but it uses a proper generative model that includes appropriate variances, so a Factor Analyzer (FA) assigns a correctly normalized density to every point in intensity space. This makes it possible to smoothly blend together the individual distributions of multiple FA's when modeling a smooth non-linear manifold. It also makes it possible to compute the posterior probabilities that are required in the E step of the EM algorithm, so the hard assignments used in K-means can be replaced by soft assignments when fitting a mixture of FA's to a set of images.

Once the manifold for each digit class has been accurately modeled by a mixture of FA's, it is possible to discriminate pre-segmented digits by seeing which mixture assigns the highest density to the image (Hinton, Dayan and Revow, 1997). But it is also possible to perform much more ambitious tasks without any further learning. If the image contains several unsegmented digits, we can relax the constraint that the data should be explained by exactly one of the mixtures and we can then ask for the best explanation of the image in terms of multiple digit instances. Finding this explanation requires correct Bayesian inference but it does not require any segmentation heuristics.

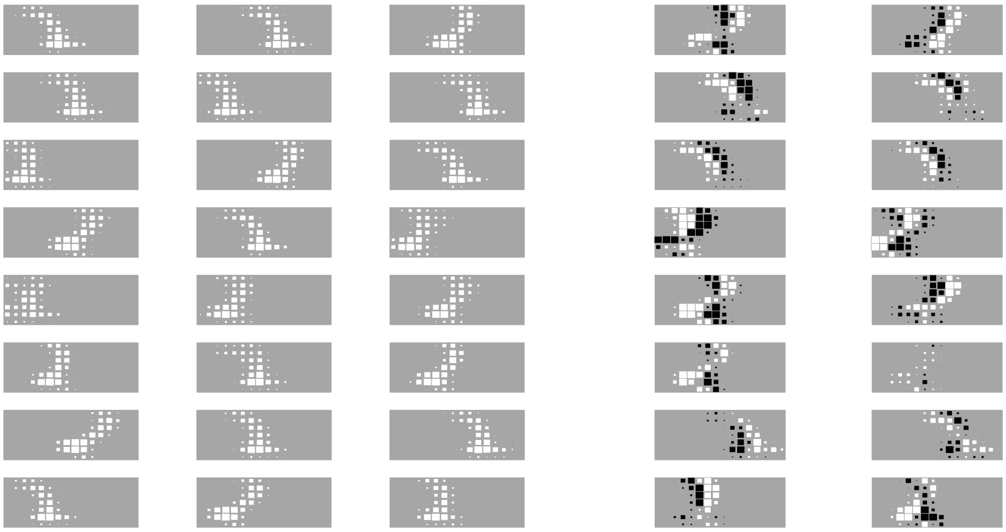
2 The models of individual digits

We first use images containing only one digit to learn four separate mixture models for the digits 2,3,4,5. The images are generated from the elastic digit models described in (Revow, Williams and Hinton, 1996) on 7×16 arrays with each image occupying roughly a 5×5 box. For each class, the deformation is allowed to vary randomly over a limited range and the only other variations are in orientation (about 28 degrees variation about the vertical) and in horizontal position (11 pixels of translation). For each class we use a mixture of 25 factor analyzers. Each FA has 2 factors and all the FA's within one mixture use the same noise model for the pixels. This noise model is learned and uses different noise levels for each pixel. Each mixture is trained on 200 digit images, plus 20 blank images to encourage it to use at least one FA to represent the absence of a digit of that class.

During training up to 100 EM iterations are allowed, but training stops if the change in the $\log_e$ probability of the data between subsequent steps is less than 10^{-5}. After learning, the means of the FA's for one digit and the factor loadings of a few of the FA's are shown in figure 1.

3 Perceptual inference using multiple mixtures of factor analyzers

We generate a 3 digit string that is hard to segment in the following way: First, we decide on the digit-class of each of the three digits, making sure they are all from different classes. Then we use the appropriate elastic model to generate an image



(a) (b)

Figure 1: (a) Means of the 24 non-blank analyzers in the mixture for twos. (b) Factor loading matrices for the 8 analyzers in the rightmost column of (a).

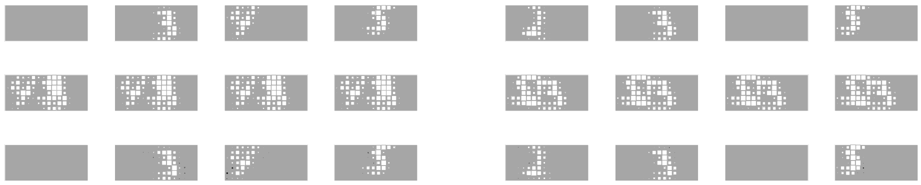
of the first digit in the string, making sure that its center lies in the left third of the image. Next we generate an image of the second digit with its center in the middle third of the image. Finally, we generate the third digit to lie in the right third of the image. The images of the three digits are then superimposed. Images produced in this way contain at most a single instance of each digit class and the spatial relationship between neighbouring digits varies from large to mild degrees of overlap providing a good test of the system's ability to segment.

To perform the segmentation using the four mixture models, we repeatedly present each mixture model with the residual image that is obtained by subtracting the current predictions of the three other mixture models. Each FA within the mixture model then computes a probability density for the residual data. Using these likelihoods and the learned mixing proportions of the FA's, one of the FA's is chosen at random. The factor activations of this FA are also randomly chosen from their Gaussian posterior distribution. The mean image predicted by these factor activations is then the new prediction of that mixture model.

When a residual image is presented to one of the mixtures, we use a noise model that is obtained by adding together the variances of the four separate noise models for the four mixtures. This allows for the fact that the "data" in the residual image is uncertain because each of the other three mixtures makes noisy predictions.

If one of the mixture models grabs the wrong part of the image early on, it is hard for the system to recover. We therefore use a form of simulated annealing as an outer loop to facilitate the escape from poor local minima. The temperature follows a flipped sigmoid, suitably offset so that it decays from a temperature of 75 to a temperature of 1 in 15 iterations.

A test set of 100 digit strings was generated as described above. After examining the



(a) (b)

Figure 2: Illustration of the system performing segmentation. The top row shows the 3 images which make up the composite image shown four times in the middle row. The bottom row shows the reconstructed images from each of the 4 mixtures. Each mixture has the option of choosing the FA within the mixture that represents the blank image. (a) Example when human and system both correctly segmented the image. (b) Example in which human incorrectly identified the string as 543.

segmented training digits, M.R. viewed hard copies of the test images and recorded his opinion of the digit sequence. He made 29 errors. When the system was run on the 100 test images, it made 21 errors[1]. Examples of test images and the resulting segmentations are shown in figure 2. By allowing each mixture to be used several times, we can segment strings containing several instances of the same digit-class.

Our demonstration of segmentation has two major weaknesses: First, the spatial relationships between digits within a string are not modeled. Second, the number of FA's required within each mixture scales as n^k, where k is the number of dimensions of variation that have non-linear effects on pixel intensities, and n is the number of different linear regimes along each dimension. Factor analysis can be modified to fix the first problem and the modification allows a hierarchical solution to the second problem.

In a mixture of FA's, the posterior probability distribution across FA's and the factor activations within each analyzer convey information about the instantiation parameters of the digit. Unfortunately, this information is scattered across the analyzers which means that later stages of processing require a lot of connections to see the information. These connections all have weights that need to be learned. It would be much more convenient if the information was gathered together into one place, regardless of which analyzer within the mixture had responsibility for the current image. The whole mixture would then be able to communicate with higher levels using just a few numbers. Once this has been done, the instantiation parameters extracted by different mixtures can be treated as data and the next level of processing can use a mixture of FA's to capture the redundancies between the instantiation parameters of digits within a string. The hidden factors at this next level might represent the instantiation parameters of the whole string including, for example, a factor representing the average spacing between neighboring digits. This suggests a multilevel hierarchical system in which each level uses mixtures to allow instantiation parameters at one level to be non-linearly related to instantiation parameters at the next level, and the information in all the FA's within a mixture is gathered together before being passed to the next level up. A hierarchy seems essential to capture the fact that the relationships between digits are generally looser than the relationships between the parts of one digit, and the parts are in

[1]To recover the order of the digits in the string, the system had to estimate the translation instantiation parameter of each digit as described in section 4.

turn, more loosely related than the edges that compose a piece of a stroke.

4 Extracting a Compact Representation of Instantiation Parameters

We start by presenting a supervised learning procedure for extracting instantiation parameters and then describe how the supervision can be eliminated. Factor analysis is a density model. But it can be modified to produce a *conditional* density model that models the probability distribution of pixel intensities, $p(\mathbf{z})$ *given* the instantiation parameters, $\mathbf{x}$, of the digit.

$$p(\mathbf{z}) = \sum_i \pi_{i|\mathbf{x}} \int p(\mathbf{y}_i|\mathbf{x})p(\mathbf{z}|\mathbf{y}_i)d\mathbf{y}_i \tag{1}$$

where i is an index over FA's and the mixing proportion ($\pi_{i|\mathbf{x}}$) of each FA depends on the distance between $\mathbf{x}$ and the FA's "center" in the space of instantiation parameters:

$$\pi_{i|\mathbf{x}} = \frac{\exp(-||\mathbf{x} - \boldsymbol{\mu}_i||^2/\sigma^2)}{\sum_j \exp(-||\mathbf{x} - \boldsymbol{\mu}_j||^2/\sigma^2)} \tag{2}$$

The instantiation parameters also determine the expected distribution of the factor activations in the selected FA via a weight matrix $\mathbf{W}$. Gaussian noise with covariance Σ is added to give:

$$p(\mathbf{y}_i|\mathbf{x}) = (2\pi)^{-k/2} \mid \Sigma \mid^{-1/2} \exp\left(-[\mathbf{y}_i - \mathbf{W}_i(\mathbf{x} - \boldsymbol{\mu}_i)]'\Sigma^{-1}[\mathbf{y}_i - \mathbf{W}_i(\mathbf{x} - \boldsymbol{\mu}_i)]\right) \tag{3}$$

Since we generated the images using an elastic model, we know the true instantiation parameters, $\mathbf{x}$, for each digit image. Given $\mathbf{x}$, the model can be fitted using the following algorithm. The mixture of FA's is fitted unsupervised as described above. Using current values for $\mathbf{W}$ $\boldsymbol{\mu}$ and Σ, one FA is picked from the posterior distribution across the mixture of FA's and values of the factors of the chosen FA are also picked from their posterior distribution. We are now in a position to do Gibbs sampling at the level of the instantiation parameters $\mathbf{x}$ (see below). Using these samples of $\mathbf{x}$ and $\mathbf{y}$ we perform an M-step to compute updated values of $\mathbf{W}$, $\mathbf{y}$ and Σ.

After fitting a separate mixture of conditional factor analyzers to each digit class, we can perform segmentation as before except that we must now perform Gibbs sampling to pick $\mathbf{x}$ from its conditional distribution given the factor activations, $\mathbf{y}_i$ of the chosen analyzer within the mixture. Eq. 3 imposes a quadratic energy function on $\mathbf{x}$ but Eq. 2 imposes a non-quadratic energy. Because of the normalization term, it is the *relative* squared distance of $\mathbf{x}$ from $\boldsymbol{\mu}_i$ that determines the energy. In practice, we simply ignore the normalization term when performing the Gibbs sampling.

Figure 3 shows that $\mathbf{x}$ can be recovered quite accurately from images of single digits as can also be recovered quite well from images of overlapping digits, with occasional large errors when the image is mis-segmented.

5 Using an autoencoder to extract compact representations

Ideally, we would like to recover $\mathbf{x}$ from images without requiring the training data to be labeled with $\mathbf{x}$. It is easy enough to fit an unconditional mixture of FA's, and

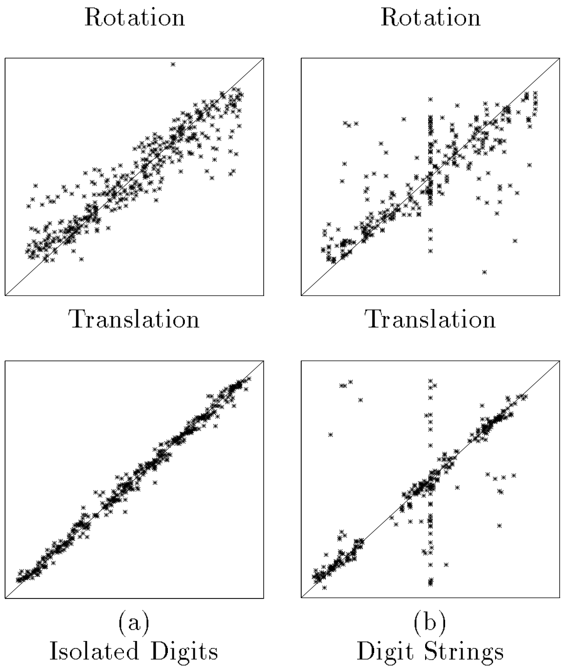


Figure 3: Illustration of the recovery of pose. The true pose parameters are plotted on the horizontal axes while the recovered values are plotted on the vertical axis. Rotations are shown in the top row and translations in the bottom row. The left panel is for isolated digits while the right panel shows recovery from strings of digits. Points lying far from the line of unity correspond to segmentation errors.

given the factor activations, $\mathbf{y}_i$ and posterior probabilities, r_i, of these analyzers, $\mathbf{x}$ can be estimated by a linear model of the form

$$\hat{\mathbf{x}} = \sum_i r_i(\boldsymbol{\mu}_i + \mathbf{W}_i^{-1}\mathbf{y}_i) \tag{4}$$

If $\mathbf{W}$ and $\boldsymbol{\mu}$ could be learned for all the analyzers without supervision it would be possible to fit a hierarchical model to data one level at a time starting at the bottom and working upwards.

One approach to solving this problem is to use backpropagation in a network resembling an autoencoder. The inputs to the network are the responsibilities of each FA and the factor activations weighted by the responsibilities. The activations in the hidden, bottleneck layer represent $\hat{\mathbf{x}}$, computed from the inputs using Eq. 4, where $\boldsymbol{\mu}_i$ and $\mathbf{W}_i^{-1}$ are learned input-to-hidden weights. The desired outputs are the factor activations and the responsibilities. For the factor activations, the squared errors are weighted by the responsibility of the relevant FA before backpropagating so the network does not need to reconstruct the activations for unused FA's. For the responsibilities, we use a cross-entropy error function $E = \sum_i r_i \log \hat{r}_i$ and the estimated responsibility is given by:

$$\hat{r}_i = \frac{\exp(-||\mathbf{x} - \boldsymbol{\mu}_i||^2/\sigma^2)}{\sum_j \exp(-||\mathbf{x} - \boldsymbol{\mu}_j||^2/\sigma^2)} \tag{5}$$

Unfortunately, this autoencoder tends to get jammed in local optima. Each FA grabs a different region of the $\mathbf{x}$ space by using a different $\boldsymbol{\mu}$. Each FA also represents a linear patch of the low-dimensional, non-linear manifold in pixel intensity space. But the ordering of the regions in $\mathbf{x}$ space does not necessarily correspond to the ordering of the patches on the manifold, and it is hard for regions to jump over one another without increasing the reconstruction errors of the autoencoder.

Fortunately, the same error function can be used to learn a generative model that does not get jammed because it allows FA's to partially commit themselves to several widely separated regions of $\mathbf{x}$ space early in the learning and then to give up on the regions that do not agree with their neighbors. In the generative model, $\mathbf{x}$ is chosen randomly. It then causes a pattern of activation in a large set of units that have fixed, Gaussian receptive fields in $\mathbf{x}$ space. These RBF units then provide inputs to groups of linear units that are used to reconstruct the factor activations and a group of softmax units that are used to reconstruct the responsibilities of the FA's. If there is noise in the RBF units, this generative model resembles the one used in Zemel and Hinton (1995). If the RBF units are noise-free and the posterior distribution over $\mathbf{x}$ when given a data vector is approximated by a set of delta functions at finely spaced grid points in $\mathbf{x}$ space, the model resembles GTM (Bishop, Svensen and Williams, 1997), except that a gradient learning algorithm must be used for the weights that reconstruct responsibilities.

We are currently applying the GTM-like version of this generative model to the task of extracting a compact representation of the instantiation parameters of a digit from the outputs of a mixture of FA's. We anticipate that it will work much better than applying GTM directly to the pixel intensities. We also anticipate that it will allow bottom-up learning of a hierarchical system in which the digits themselves are composed of features whose instantiation parameters are extracted from local image patches. These results will be reported in the final version of the paper.

Acknowledgements

We thank Zoubin Ghahramani and Peter Dayan. This research was funded IRIS and NSERC. Hinton is the Nesbitt-Burns fellow of CIAR.

References

[1] C. M. Bishop, M. Svensen, and C. K. I. Williams. GTM: The generative topographic mapping. *Neural computation*, 1997.

[2] G. E. Hinton, P. Dayan, and M. Revow. Modeling the manifolds of images of handwritten digits. *IEEE Transactions Neural Networks*, (8):65–74, 1997.

[3] M. Revow, C. K. I. Williams, and G. E. Hinton. Using generative models for handwritten digit recognition. *IEEE Transactions Pattern Analysis and Machine Intellegince*, 18(6):592–606, 1996.

[4] P. Simard, Y. Le Cun, and J. Denker. Efficient pattern recognition using a new transformation distance. In J. D. Cowan S. J. Hanson and C. L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 50–58. Morgan Kaufmann, 1993.

[5] R. S. Zemel and G. E. Hinton. Learning population codes by minimizing description length. *Neural computation*, 7(3):549–564, 1995.