Empirical Entropy Manipulation and Analysis

Paul Viola, Nicol N. Schraudolph, Terrence J. Sejnowski
Computational Neurobiology Laboratory
The Salk Institute for Biological Studies
10010 North Torrey Pines Road
La Jolla, CA 92037-1099
viola@salk.edu

Abstract

No finite sample is sufficient to determine the density, and therefore the entropy, of a signal directly. Some assumption about either the functional form of the density or about its smoothness is necessary. Both amount to a prior over the space of possible density functions. By far the most common approach is to assume that the density has a parametric form.

By contrast we derive a differential learning rule called EMMA that optimizes entropy by way of kernel density estimation. Entropy and its derivative can then be calculated by sampling from this density estimate. The resulting parameter update rule is surprisingly simple and efficient.

We will describe two real-world applications that can be solved efficiently and reliably using EMMA. In the first application EMMA is used to align 3D models to complex natural images. In the second application EMMA is used to detect and correct corruption in magnetic resonance images (MRI). Both applications are beyond the scope of existing parametric entropy models.

1 Introduction

Information theory is playing an increasing role in unsupervised learning and visual processing. For example, Linsker has used the concept of information maximization to produce theories of development in the visual cortex (Linsker, 1988). Becker and Hinton have used information theory to motivate algorithms for visual processing (Becker and Hinton, 1992). Bell and Sejnowski have used information maximization to solve the "cocktail party" or signal separation problem (Bell and Sejnowski, 1995). In order to simplify analysis and implementation, each of these techniques makes specific assumptions about the nature of the signals used, typically that the signals are drawn from some parametric density. In practice, such assumptions are very inflexible.

In this paper we will derive a procedure that can effectively estimate and manipulate the entropy of a wide variety of signals using non-parametric densities. Our technique is distinguished by is simplicity, flexibility and efficiency.

We will begin with a discussion of principal components analysis (PCA) as an example of a simple parametric entropy manipulation technique. After pointing out some of PCA's limitation, we will then derive a more powerful non-parametric entropy manipulation procedure. More significantly, we will show that the same entropy estimation procedure can be used to tackle several difficult visual processing problems. Finally, we will demonstrate the strength of this method on two real-world applications.

1.1 Parametric Entropy Estimation

Typically parametric entropy estimation is a two step process. We are given a parametric model for the density of a signal and a sample. First, from the space of possible density functions the most probable is selected. This often requires a search through parameter space. Second, the entropy of the most likely density function is evaluated.

Parametric techniques can work well when the assumed form of the density matches the actual data. Conversely, when the parametric assumption is violated the resulting algorithms are incorrect. The most common assumption, that the data follow the Gaussian density, is especially restrictive. An entropy maximization technique that assumes that data is Gaussian, but operates on data drawn from a non-Gaussian density, may in fact end up minimizing entropy.

The popularity of the Gaussian is based on three considerations: (1) finding the Gaussian that fits the data best is very easy, (2) the entropy of the Gaussian can be directly calculated from its variance, and (3) an affine transformation of a Gaussian random variable remains Gaussian. The entropy of a Gaussian density is

$$h(X) = -E_X[\log g_\psi(x-\mu)] = \frac{1}{2}\log 2e\pi\psi$$

where $g_\psi(x-\mu)$ is the Gaussian density with variance ψ and mean μ and $E_X[\cdot]$ is the expectation over the random variable X. It is well known that given a sample A, the most likely Gaussian density has as its mean the mean of A and as its variance the variance of A. As a result, if we assume that a random variable is Gaussian, its empirical entropy is proportional to the log of sample variance. More simply, when the data is a assumed Gaussian, maximizing entropy is equivalent to maximizing variance.

1.2 Example: Principal Components Analysis

There are a number of signal processing and learning problems that can be formulated as entropy maximization problems. One prominent example is *principal component analysis* (PCA). Given a random variable X, a vector v can be used to define a new random variable, $Y_v = X \cdot v$ with variance $\mathrm{Var}(Y_v) = E_X[(X \cdot v - E_X(X \cdot v))^2]$. The principal component $\hat{v}$ is the unit vector for which $\mathrm{Var}(Y_{\hat{v}})$ is maximized.

In practice neither the density of X nor Y_v is known. The projection variance is computed from a sample A of points from X,

$$\mathrm{Var}(Y_v) \approx \underset{A}{\mathrm{Var}}(Y_v) \equiv E_A[(X \cdot v - E_A[X \cdot v])^2] \ , \tag{1}$$

where $\mathrm{Var}_A(Y_v)$ and $E_A[\cdot]$ are shorthand for variance and mean evaluated over the sample A. Oja has derived an elegant on-line rule for learning $\hat{v}$ when presented with a sample of X (Oja, 1982).

Under the assumption that X is Gaussian is is easily proven that $Y_{\hat{v}}$ has maximum entropy. Moreover, in the absence of noise $Y_{\hat{v}}$ contains maximal information about X. However, when X is *not* Gaussian $Y_{\hat{v}}$ is generally not the most informative projection.

2 Estimating Entropy with Parzen Densities

We will now derive a general procedure for manipulating and estimating the entropy of a random variable from a sample. Given a sample of a random variable X, we can construct another random variable $Y = F(X, v)$. The entropy, $h(Y)$, is a function of v and can be manipulated by changing v. Since there is no direct technique for finding the parameters that will extremize $h(Y)$ we will search the parameter space using gradient descent. The derivation assumes that Y is a vector random variable. The joint entropy of a two random variables, $h(W_1, W_2)$, can be evaluated by constructing the vector random variable, $Y = [W_1, W_2]^T$ and evaluating $h(Y)$.

Rather than assume that the density has a parametric form, whose parameters are selected using maximum likelihood estimation, we will instead use Parzen window density estimation (Duda and Hart, 1973). In the context of entropy estimation, the Parzen estimate has three significant advantages over maximum likelihood: (1) it can model the density of any signal provided the density function is smooth; (2) since the Parzen estimate is computed directly from the sample, there is no search for parameters; (3) the derivative of the entropy of the Parzen estimate is simple to compute.

The form of the Parzen estimate constructed from a sample A is

$$P^*(y, a) = \frac{1}{N_A} \sum_{y_A \in a} R(y - y_A) = E_A[R(y - y_A)] \quad , \tag{2}$$

where the Parzen estimator is constructed with the window function $R(\cdot)$ which integrates to 1. The Parzen density is an unbiased estimate for the density of a signal perturbed by random noise with density $R(\cdot)$. In our subsequent analysis we will assume that the Parzen window function is a Gaussian density function. This will simplify some of our subsequent analysis, but it is *not* necessary. Any differentiable function could be used. Another good choice is the Cauchy density.

Unfortunately evaluating the entropy integral

$$h(Y) \approx -E_Y[\log P^*(Y, A)] = -\int_{-\infty}^{\infty} \log P^*(y, a) dy$$

is inordinately difficult. This integral can however be approximated as a sample mean:

$$h(Y) \approx h^*(Y) \equiv -E_B[\log P^*(Y, A)] \tag{3}$$

$$= -E_B[\log E_A[R(y_A - y_B)]] \tag{4}$$

where $E_B[\,]$ is the sample mean taken over the sample B. The sample mean converges toward the true expectation at a rate proportional to $1/\sqrt{N_B}$ (N_B is the size of B). To reiterate, two samples can be used to estimate the entropy of a density: the first is used to estimate the density, the second is used to estimate the entropy[1]. We call $h^*(Y)$ the EMMA estimate of entropy[2].

One way to extremize entropy is to use the derivative of entropy with respect to v. This may be expressed as

$$\frac{d}{dv} h(Y) \approx \frac{d}{dv} h^*(Y) = -\frac{1}{N_B} \sum_{y_B \in B} \frac{\sum_{y_A \in A} \frac{d}{dv} g_\psi(y_B - y_A)}{\sum_{y_A \in A} g_\psi(y_B - y_A)} \tag{5}$$

[1] Using a procedure akin to leave-one-out cross-validation a single sample can be used for both purposes.

[2] EMMA is a random but pronounceable subset of the letters in the words "Empirical entropy Manipulation and Analysis".

$$= -\frac{1}{N_B} \sum_{y_B \in B} \sum_{y_A \in A} W_y(y_B, y_A) \frac{d}{dv} \frac{1}{2} D_\psi(y_B - y_A) \ , \tag{6}$$

where

$$W_y(y_1, y_2) \equiv \frac{g_\psi(y_1 - y_2)}{\sum_{y_A \in A} g_\psi(y_1 - y_A)} \ , \tag{7}$$

$$D_\psi(y) \equiv y^T \psi^{-1} y$$

and $g_\psi(y)$ is a multi-dimensional Gaussian with covariance ψ. $W_y(y_1, y_2)$ is an indicator of the degree of match between its arguments, in a "soft" sense. It will approach one if y_1 is significantly closer to y_2 than any element of A. To reduce entropy the parameters v are adjusted such that there is a reduction in the average squared distance between points which W_y indicates are nearby.

2.1 Stochastic Maximization Algorithm

Both the calculation of the EMMA entropy estimate and its derivative involve a double summation. One summation is over the points in sample A and another over the points in B. As a result the cost of evaluation is quadratic in sample size: $O(N_A N_B)$. While an accurate estimate of empirical entropy could be obtained by exhaustively sampling the data, a stochastic estimate of the entropy can be obtained with much less computation. This is especially critical in entropy manipulation problems, where the derivative of entropy is evaluated many hundreds or thousands of times. Without the quadratic savings that arise from using smaller samples entropy manipulation would be impossible.

We have proven that a gradient ascent procedure using a stochastic version of EMMA will converge to solutions that are near maximum (or minimum) of entropy (Viola, 1995). The proof assumes that the Parzen estimate will converge to the true density.

2.2 Estimating the Covariance

In addition to the learning rate λ, the covariance matrices of the Parzen window functions densities g_ψ are important parameters of EMMA. These parameters may be chosen so that they are optimal in the maximum likelihood sense. For simplicity, we assume that the covariance matrices are diagonal, $\psi = \text{DIAG}(\sigma_1^2, \sigma_2^2, \ldots)$. Following a derivation almost identical to the one described in Section 2 we can derive an equation analogous to (5),

$$\frac{d}{d\sigma_k} h^*(Y) = -\frac{1}{N_B} \sum_{y_B \in b} \sum_{y_A \in a} W_y(y_B, y_A) \left(\frac{1}{\sigma_k}\right) \left(\frac{[y]_k^2}{\sigma_k^2} - 1\right) \tag{8}$$

where $[y]_k$ is the kth component of the vector y. The optimal, or most likely, ψ minimizes $h^*(Y)$. In practice both v and ψ are adjusted simultaneously; for example, while v is adjusted to maximize $h^*(Y_v)$, ψ is adjusted to minimize $h^*(Y_v)$.

3 Principal Components Analysis and Information

As a demonstration, we can derive a parameter estimation rule akin to principal components analysis that truly maximizes information. This new EMMA based component analysis (ECA) manipulates the entropy of the random variable $Y_v = X \cdot v$ under the constraint that $|v| = 1$. For any given value of v the entropy of Y_v can be estimated from a sample of X as:

$$h^*(Y_v) = \frac{1}{N_B} \sum_{x_B \in B} log \left(\frac{1}{N_A} \sum_{x_A \in A} g_\psi(x_B \cdot v - x_A \cdot v) \right)$$

where ψ is the variance of the Parzen smoothing function. Moreover we can estimate the derivative of entropy:

$$\frac{d}{dv}h^*(Y_v) = \frac{1}{N_B}\sum_B\sum_A W_y(y_B, y_A)\,\psi^{-1}(y_B - y_A)(x_B - x_A) \ ,$$

where $y_A = x_A \cdot v$ and $y_B = x_B \cdot v$. The derivative can be decomposed into parts which can be understood more easily. Ignoring the weighting function $W_y\psi^{-1}$ we are left with the derivative of some unknown function $f(Y_v)$:

$$\frac{d}{dv}f(Y_v) = \sum_B\sum_A(y_B - y_A)(x_B - x_A) \tag{9}$$

$$= N_B N_A E_B[E_A[(y_B - y_A)(x_B - x_A)]] \ . \tag{10}$$

What then is $f(Y_v)$? The derivative of the squared difference between samples is:

$$\frac{d}{dv}(y_B - y_A)^2 = 2(y_B - y_A)(x_B - x_A) \ .$$

So we can see that

$$f(Y_v) = N_B N_A E_B[E_A[(y_B - y_A)^2]]$$

is the expectation of the squared difference between pairs of trials of Y_v.

Recall that PCA searches for the projection, Y_v, that has the largest sample variance: $\mathrm{Var}_A(Y_v) = E_A[(y_A - E_A[y_A])^2]$. Interestingly, $f(Y_v)$ is precisely twice the sample variance. Without the weighting term $W_y\psi^{-1}$, ECA would find exactly the same vector that PCA does: the maximum variance projection vector. However because of W_y the derivative of ECA does not act on all points of Y_v equally. Points that are very far apart are forced no further apart. Another way of interpreting (ECA) is as a type of robust variance maximization. Points that might best be interpreted as outliers, because they are very far from the body of other points, play a very small role in the minimization. This robust nature stand in contrast to PCA which is very sensitive to outliers.

For densities that are Gaussian, the maximum entropy projection is the first principal component. In simulations ECA effectively finds the same projection as PCA, and it does so with speeds that are comparable to Oja's rule. ECA can be used both to find the entropy maximizing (ECA-MAX) and minimizing (ECA-MIN) axes. For more complex densities the PCA axis is very different from the entropy maximizing axis. To provide some intuition regarding the behavior of ECA we have run ECA-MAX, ECA-MIN, Oja's rule, and two related procedures, BCM and BINGO, on the same density. BCM is a learning rule that was originally proposed to explain development of receptive fields patterns in visual cortex (Bienenstock, Cooper and Munro, 1982). More recently it has been argued that the rule finds projections that are far from Gaussian (Intrator and Cooper, 1992). Under a limited set of conditions this equivalent to finding the minimum entropy projection. BINGO was proposed to find axes along which there is a bimodal distribution (Schraudolph and Sejnowski, 1993).

Figure 1 displays a 400 point sample and the projection axes discussed above. The density is a mixture of two clusters. Each cluster has high kurtosis in the horizontal direction. The oblique axis projects the data so that it is most uniform and hence has the highest entropy; ECA-MAX finds this axis. Along the vertical axis the data is clustered and has low entropy; ECA-MIN finds this axis. The vertical axis also has the highest variance. Contrary to published accounts, the first principal component can in fact correspond to the *minimum* entropy projection. BCM, while it may find minimum entropy projections for some densities, is attracted to the kurtosis along the horizontal axis. For this distribution BCM *neither* minimizes nor maximizes entropy. Finally, BINGO successfully discovers that the vertical axis is very bimodal.

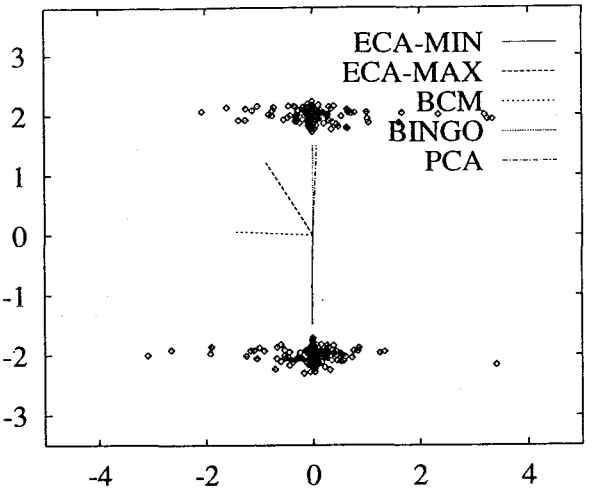

Figure 1: At left: A scatter plot of a 400 point sample from a two dimensional density. Included are the output of PCA (vertical axis), ECA-MAX (oblique axis), ECA-MIN (vertical), BCM (horizontal) and BINGO (vertical).

4 Applications

EMMA has proven useful in a number of applications. We will briefly describe two here. The first application finds the alignment between a three dimensional model and an image using mutual information. While this problem has been of interest for over 30 years, progress has been hampered by the sheer complexity of the relationship between an object and its image, which involves the object's shape, surface properties, position, and illumination. For example, changes in illumination can radically alter the intensity and shading of an image. Though the human visual system can use shading both for recognition and image interpretation, most existing computer vision systems cannot.

4.1 3-D Model Alignment

The mutual information between object normals and image intensities can be used to evaluate the alignment of a 3D object and an image. In general mutual information is closely related to predictability. By maximizing mutual information we are choosing an alignment that makes the image most easily predictable from the model. More concretely, an alignment associates the intensities from the image with points on the surface of the object[3]. For most realistic images there is a functional relationship between the surface normal at a point and the observed intensity. This relationship is determined by the surface reflectance and the illumination, it is known as a reflectance map. Even though there may be a different reflectance map for each object and image, there will be mutual information between the normals of the model and the intensities of the image. As a result, the mutual information is insensitive to changes in illumination and surface properties.

This approach is unique in that it compares 3D object models directly to raw images; no pre-processing or edge detection is required. Using EMMA a gradient ascent alignment procedure can be defined that adjusts object pose until the mutual information between image and object is maximized. EMMA based alignment is surprisingly efficient, requiring between 10 and 60

[3]This is much like the texture mapping operation from computer graphics.

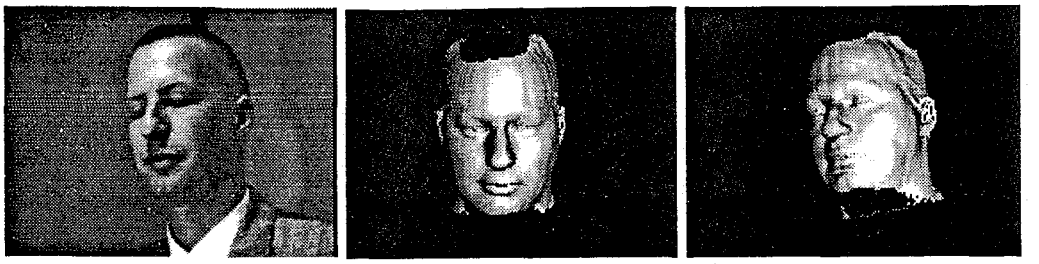

Figure 2: Left: A typical image. Center: A rendering of a model derived from a cyberware scan in an incorrect pose. Right: The same model in the pose obtained by optimizing mutual information. The optimization proceeds over the space of 3D rotations and translations.

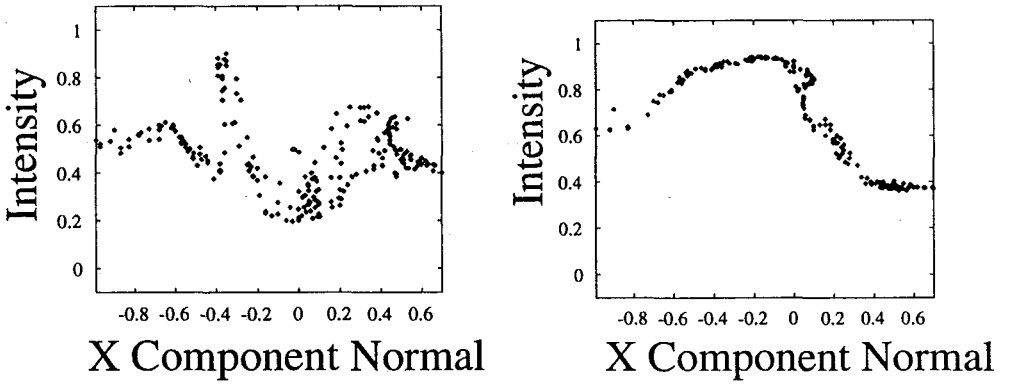

Figure 3: A scatter plot the intensity of the video image versus the x component of the surface normal from the model (a single value of the y component is used). At left the image and model are misaligned. At right they are aligned.

seconds on a Sparc 10. The same approach has been applied to a number of different alignment problems (see (Viola and Wells, 1995) for a complete description).

Figure 2 shows an example image and renderings of two different poses for a 3D model. The rendered images, which have been constructed using a model for lighting and reflectance, are included for visualization only; they are not used as part of the algorithm. Figure 3 shows part of the joint density of normals and intensities for two different alignments. While these distributions would be difficult to model parametrically, EMMA can estimate mutual information and its derivative. We are aware of no other technique that can effectively solve this problem.

4.2 MRI Processing

In the second application EMMA is used to process magnetic resonance images (MRI). An MRI is a 2 or 3 dimensional image that records the density of tissues inside the body. In the head, as in other parts of the body, there are a number of distinct tissue classes including: bone, water, white matter, grey matter, and fat. In principle the density of pixel values in an MRI should be clustered, with one cluster for each tissue class. In reality MRI signals are corrupted by a bias field, a multiplicative offset that varies slowly in space. The bias field results from unavoidable variations in magnetic field (see (Wells III et al., 1994) for an overview of this problem).

Because of clustering an uncorrupted MRI should have relatively low entropy. Corruption from

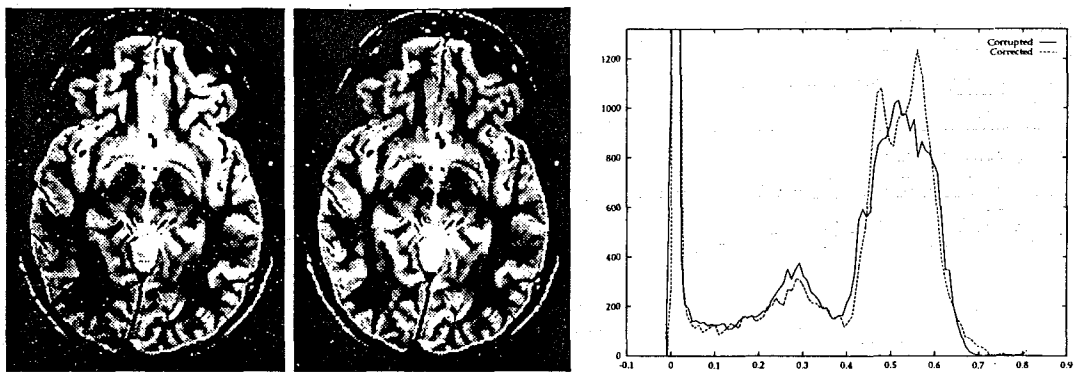

Figure 4: At left: A slice from an MRI scan of a head. Center: The scan after correction. Right: The density of pixel values in the MRI scan before and after correction.

the bias field perturbs the MRI image, increasing the values of some pixels and decreasing others. The bias field acts like noise, adding entropy to the pixel density. We use EMMA to find a low-frequency *correction* field that when applied to the image, makes the pixel density have a lower entropy. The resulting corrected image will have a tighter clustering than the original density.

Figure 4 shows an MRI scan and a histogram of pixel intensity before and after correction. The difference between the two scans is quite subtle: the uncorrected scan is brighter at top right and dimmer at bottom left. This non-homogeneity makes constructing automatic tissue classifiers difficult. In the histogram of the original scan white and grey matter tissue classes are confounded into a single peak ranging from about 0.4 to 0.6. The histogram of the corrected scan shows much better separation between these two classes. For images like this the correction field takes between 20 and 200 seconds to compute on a Sparc 10.

5 Conclusion

We have demonstrated a novel entropy manipulation technique working on problems of significant complexity and practical importance. Because it is based on non-parametric density estimation it is quite flexible, requiring no strong assumptions about the nature of signals. The technique is widely applicable to problems in signal processing, vision and unsupervised learning. The resulting algorithms are computationally efficient.

References

Anderson, J. and Rosenfeld, E., editors (1988). *Neurocomputing: Foundations of Research.* MIT Press, Cambridge.

Becker, S. and Hinton, G. E. (1992). A self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355:161–163.

Bell, A. J. and Sejnowski, T. J. (1995). An information-maximisation approach to blind separation. In *Advances in Neural Information Processing*, volume 7, Denver 1994. Morgan Kaufmann, San Francisco.

Bienenstock, E., Cooper, L., and Munro, P. (1982). Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, 2. Reprinted in (Anderson and Rosenfeld, 1988).

Duda, R. and Hart, P. (1973). *Pattern Classification and Scene Analysis.* Wiley, New York.

Intrator, N. and Cooper, L. N. (1992). Objective function formulation of the bcm theory of visual cortical plasticity: Statistical connections, stability conditions. *Neural Networks*, 5:3–17.

Linsker, R. (1988). Self-organization in a perceptual network. *IEEE Computer*, pages 105–117.

Oja, E. (1982). A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15:267–273.

Schraudolph, N. N. and Sejnowski, T. J. (1993). Unsupervised discrimination of clustered data via optimization of binary information gain. In Hanson, S. J., Cowan, J. D., and Giles, C. L., editors, *Advances in Neural Information Processing*, volume 5, pages 499–506, Denver 1992. Morgan Kaufmann, San Mateo.

Viola, P. A. (1995). *Alignment by Maximization of Mutual Information.* PhD thesis, Massachusetts Institute of Technology.

Viola, P. A. and Wells, W. (1995). Alignment by maximization of mutual information. In *Proceedings of the International Conference on Computer Vision*, Cambridge, MA. IEEE, Washington, DC.

Wells III, W., Grimson, W., Kikinis, R., and Jolesz, F. (1994). Statistical Gain Correction and Segmentation of MRI Data. In *Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition*, Seattle, Wash. IEEE , *Submitted.*