Improved Generative Models for Continuous Image Features through Tree-structured Non-parametric Distributions

Marwan A. Mattar **Erik G. Learned-Miller**

Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003

{mmattar, elm}@cs.umass.edu

Abstract

Density estimation arises in a wide range of vision problems and methods which can deal with high dimensional image features are of great importance. While in principle a non-parametric distribution can be estimated for the full feature distribution using Parzen windows technique, the amount of data to make these estimates accurate is usually either unattainable or unmanageable. Consequently, most modelers resort to parametric models such as mixtures of Gaussians (or other more complicated parametric forms) or make independence assumptions about the features. Such assumptions could be detrimental to the performance of vision systems since realistically, image features have neither a simple parametric form, nor are they independent.

In this paper, we revive non-parametric models for image feature distributions by finding the best tree-structured graphical model (using the Chow-Liu algorithm) for our data, and estimating non-parametric distributions over the one- and two-node marginals necessary to define the graph. This procedure has the appealing property that, if the tree-structured model represents the true conditional independence relations for the features, then our estimated joint distribution converges rapidly to the true distribution of the data. Even when this is not true, it converges to the best possible tree-structured model for the original distribution. We illustrate the effectiveness of this technique on simulated data and a real-world plankton classification problem.

1 Introduction

Modeling the joint distributions of *continuous* image features is a difficult task since in most applications only a relatively small number of samples are available for a high-dimensional distribution. In principle, if we had an infinite number of samples, we could estimate the true joint distribution over all the features using a kernel density estimation (KDE) technique such as Parzen windows [20, 9]. In a standard classification setting, we could use the estimated class conditional densities to build a classifier that attains the Bayes error rate. However, in reality, the required number of samples is either unattainable or unmanageable. This has led many to conclude that non-parametric techniques are not viable and are impractical for high-dimensional distributions. Consequently, most modelers resort to making simplifying assumptions about their distributions such as:

- Restricting their form to simple parametric distributions. While such assumptions reduce the variance of density estimates, they do so at the cost of higher bias, since typically, true distributions (at least for continuous image features) are not well fit by these simple parametric forms.
- Assuming the features are independent. Now the joint distribution factors into the product of one-dimensional distributions which can be estimated accurately using KDE. This may result in good

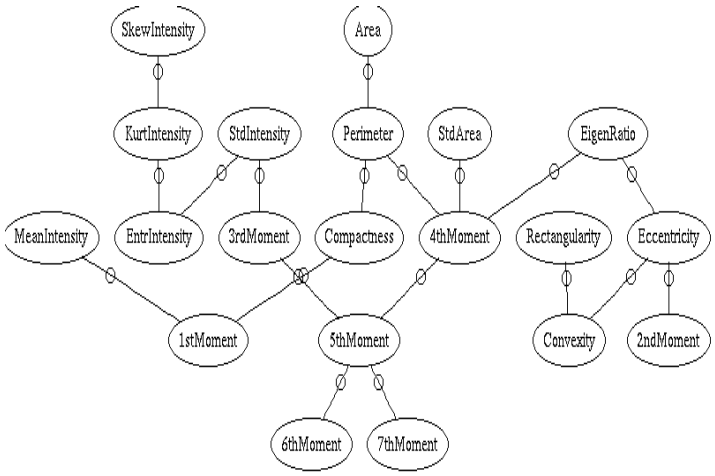


Figure 1: An example of a learned tree model. This graph shows the learned dependencies between 20 features for one of the classes in the phytoplankton data set. See Section 6.1 for a brief description of the features. The edges on the graph connect the features that have the maximum dependencies or greatest mutual information. The connections in the graph match intuitions, for the most part, about which features should have the highest mutual information, such as the edge between EigenRatio and Eccentricity and the edge between Area and Perimeter. Note furthermore that this is a graph over *continuous*, not discrete, random variables.

estimates of marginal feature distributions, but a poor model for the joint feature distribution, since the assumption of total feature independence is seldom even approximately realized.

- Discretizing the feature space. This procedure is problematic, because small changes in the bin size can have a profound effect on the observed probability mass function. A further difficulty is that if the bin width is chosen small enough to capture local structure, then, even in two dimensions, the total number of bins becomes so large that random error effects are likely to become dominant [20].

In this paper, we revive non-parametric models for image feature distributions by finding the optimal tree-structured graph (see Figure 1) using the Chow-Liu algorithm [6], and estimating non-parametric distributions over the one- and two-node marginals necessary to define the graph.

More specifically, we propose a simple procedure for effective MAP classification using tree-structured non-parametric distributions:

1. Obtain a training set of labeled data for each class.
2. For the points in each class, learn a tree structured graph that spans the strongest dependencies among the features.
3. Estimate a non-parametric, but structured, probability density for each class based upon the learned tree-structured graph for that class.
4. Evaluate the likelihood of each class by evaluating its probability density under the estimated distribution.

There are two very appealing properties of this model. First, as the amount of training data grows, our non-parametric estimate will become closer and closer (and ultimately converge) to the *best possible approximation* (in the sense of Kullback-Leibler [7] divergence) of the true distribution over all possible tree-structured distributions.[1] This includes not only parametrically structured tree distributions, but all non-parametric tree-structured distributions as well. If the tree-structured graph represents the true conditional independence relations for the features, then our estimated joint distribution converges to the true distribution of the data. Second, by restricting our attention to tree-structured distributions, each distribution can be expressed as a simple function of the one- and two-node joint distributions of the features. By limiting the factors of the joint distribution to two dimensions, we ensure that our non-parametric estimates of these factors converge rapidly, and can be reasonably estimated with practically sized data sets.

[1] This is a simple consequence of the optimality of the Chow-Liu algorithm and the statistical consistency of non-parametric density estimates.

In the following section, we describe previous versions of this model in the machine learning and geosciences communities, and explain the main differences with the one we present here. It is important to keep in mind that the main purpose of this paper is not to describe a newly developed model, but to thoroughly analyze a variant of an existing tool to reveal its wide applicability for the vision community. From our results on plankton classification (as discussed in Section 6) we conclude that estimating the joint distribution by a tree-structured graph produces better estimates than using parametric distributions. We believe that these models could be utilized more often in the field, and we provide experimental results on simulated distributions and a real-world plankton classification problem. We hope that the outcomes of these experiments convince the reader of the powerful properties that these models possess and their wide range of potential applications.

2 Related Work and Our Contributions

In their original paper, Chow and Liu [6] showed that the graphical model represented by the maximum spanning tree of the fully connected graph whose nodes are the random variables (or features) has the closest Kullback-Leibler (KL) divergence to the true joint distribution than any other tree-structured distribution. This result requires that the edge weight between any two nodes be equal to the mutual information between the features represented by those nodes. Although their proof was for discrete random variables, their argument scales to continuous random variables (see [2] for a formal proof). They used their algorithm to estimate class-conditional densities in a Bayesian classifier used to classify gray-scale images of digits.

More than 20 years later, this procedure for constructing a Bayesian classifier was named Bayesian Multi-nets by the the machine learning community [10, 11]. Friedman *et al.* [10] also introduced the Tree Augmented Network classifier, which uses the data points from all the classes to learn one tree structure for all the classes (as opposed to a potentially different graph for each class). Since then, a large number of structure learning algorithms have been proposed in the literature [12] and more recently researchers have been interested in comparing and contrasting generative and discriminative methods to structure learning (see [17] for a review). However, most of the previous work [10, 11, 12, 17, 1, 14] (with a few exceptions [2, 13]) assume a discrete feature space.

Perhaps the only notable exception is Bach and Jordan [2]. Bach and Jordan [2] proposed a generalization of independent components analysis (ICA) were instead of looking for a transform that makes the data components independent they look for a transform that makes the data components well fit by a tree-structured distribution. With respect to density estimation, their experiments on simulated data showed that transforming the distribution to better fit a graph structure and then estimating the one- and two-node marginals of their graph non-parametrically produced the best estimates.

Datcu *et al.* [13] is perhaps the first and only other work that applies the Chow-Liu algorithm on continuous image data while using kernel density estimation to estimate the one- and two-node marginals. However, their paper overlooks some of the issues involved with KDE such as bandwidth selection and the consistency of the marginals (see Section 4.2).

3 Background

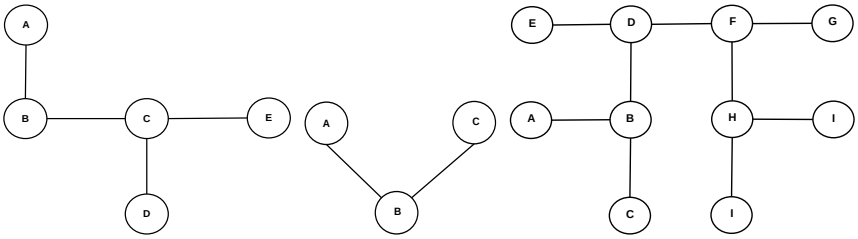
In this section we review some background material regarding naive Bayes and tree-structured distributions before describing details of our method. We begin with a labeled data set consisting of n real valued feature vectors in $\Re^d$ and their corresponding class labels. $\{c_1, c_2, ..., c_m\}$ represents the set of class labels for an m-class problem.

3.1 Naive Bayes

Naive Bayes assumes that all of the features are conditionally independent given the class, *i.e.*

$$P(\mathbf{x}|c_i) = \prod_{j=1}^{d} P(x_j|c_i). \tag{1}$$

The graphical representation of the Naive Bayes model (for the class-conditional distribution) would represent the features as nodes without any edges. This assumption produces an estimate of the distribution that has lower variance (but higher bias) than estimating the joint distribution directly. We discuss this bias/variance trade-off in more detail in Section 6. Since image features are rarely independent, the estimated density under the Naive model will tend to be an inaccurate estimate of the true underlying distribution.



(a) (b) (c)

Figure 2: Examples of tree-structured graphs. a) An example of a 5-node tree-structured graph. b) A graphical model representing the exact conditional independence relations in our synthetic 3-dimensional Gaussian distribution (see text). c) A graphical model representing an approximation to the conditional independence relations in our synthetic 10-dimensional Gaussian distribution (see text).

3.2 Tree-structured Distributions

A tree-structured distribution is one in which dependencies are represented by a tree-structured graph (see Figure 2(a)). To write the joint probability function for a tree-structured model, any vertex can be declared as the root and a directed tree can be formed by assigning arrows to point away from the root. Thus, we can define the joint probability as the product of conditional probabilities, where each term becomes a conditional distribution of a node given its parent in the directed tree. For example, the joint distribution represented by the tree graph in Figure 2(a) can be written

$$P(\mathbf{X}) = P(X_A)P(X_B|X_A)P(X_C|X_B)P(X_D|X_C)P(X_E|X_C) \quad (2)$$

$$= \frac{P(X_A,X_B)P(X_B,X_C)P(X_C,X_D)P(X_C,X_E)}{P(X_B)P(X_C)P(X_C)}. \quad (3)$$

The general form for the joint distribution on a graph is [19]

$$P(\mathbf{X}) = \frac{\prod_{\{i,j\}} P(X_i,X_j)}{\prod_k P(X_k)^{(d_k-1)}}, \quad (4)$$

where the set $\{i,j\}$ denotes the set of all the edges in the graph ($n-1$ edges), k simply iterates through all the marginals and d_k is the degree of node k.

A tree model for a set of mutually independent features is equivalent to a Naive Bayes model (through the definition of independence). For example, if all of the nodes in Figure 2(a) were mutually independent, then the tree-structured distribution, $P(\mathbf{X})$ becomes

$$P(\mathbf{X}) = \frac{P(X_A)P(X_B)P(X_B)P(X_C)P(X_C)P(X_D)P(X_C)P(X_E)}{P(X_B)P(X_C)P(X_C)} \quad (5)$$

$$= P(X_A)P(X_B)P(X_C)P(X_D)P(X_E), \quad (6)$$

a Naive Bayes model. This means that, at worst, if we model independent features, the tree estimate of the joint distribution will be as good as the Naive Bayes model. This implies that, modeling dependencies between features using a tree-structured distribution will never perform worse than a Naive Bayes model. Unless of course we have an extremely small sample size that the one-node marginals for the Naive Bayes model are estimated much better than the two-node marginals in the tree distribution.

It is important that the estimates for the one- and two-node marginals be consistent. By consistent we mean that marginalizing the joint should give us the same distribution as estimating the marginal directly from the data. If the one- and two-node marginals are inconsistent then the conditional distribution would not be a true density. Inconsistency arises if we have poor estimates of the joint and marginal distributions which usually occurs if not enough data is present. We discuss this in more detail in Section 4.2 and describe a method for enforcing consistency.

4 Algorithm

We now overview our algorithm and discuss specific details in later subsections. We perform the following two steps for each class separately, where at each iteration we only use the data points that belong to that class.

The first step computes the maximum spanning tree and the second estimates the necessary non-parametric distributions that define the graph.

Step 1 - Maximum Spanning Tree: The first stage of the algorithm is to compute the maximum spanning tree specific to the class. Before, we do that we need to compute the symmetric cost matrix between all of the nodes (*i.e.* the edge weights). As mentioned earlier, the edge weights are defined as the mutual information [7] between the features represented by the nodes. Thus, entry (i, j) of the cost matrix, C is defined as

$$C_{ij} = I(X_i; X_j) = C_{ji}, \tag{7}$$

where $I(X_i; X_j)$ is the mutual information between features X_i and X_j. Now given the cost matrix, we run one of the standard maximum spanning tree algorithms to get a subset of edges that characterize the undirected acyclic model for this class.

Step 2 - Tree Density Estimate: The next step is to estimate the class conditional density given the optimal tree structure computed in the previous step As we can see from equation (4), we need to estimate non-parametrically the 2-node marginals between the features connected by an edge and the 1-node marginals for the nodes that have a degree of 2 or higher. At test time, when we are provided with a feature vector, we compute the likelihood of its various components under the different joint and marginal density estimates and then apply equation (4) accordingly to get the likelihood of the class given that data point.

4.1 Non-parametric Density Estimation

As mentioned earlier we need to estimate the distribution from samples non-parametrically. In this section we briefly review KDE.

In a nutshell, KDE involves, placing a kernel at each sample point and then using the data to optimize the parameters for the kernels. The probability density function (PDF) is then defined as the normalized sum of all the kernels. The most appealing property of non-parametric estimates is statistical consistency. That is, as the number of sample points increases, the estimated density will become closer and closer to the true underlying distribution, eventually converging to the true distribution. Parametric and semi-parametric models do not possess this property. For example, if we are assuming that a bimodal distribution is a Gaussian, an infinite number of points will not converge the estimated Gaussian distribution to the true bimodal distribution.

When attempting to estimate a density non-parametrically, two main choices need to be made. The first is the type of kernel that will be placed at each point. In all of the experiments in this paper we use a Gaussian kernel. The second

- The first is the type of kernel that will be placed at each point which is problem-dependent. In all of the experiments in this paper we use a Gaussian kernel.
- The second is how the kernel parameters (in the case of a Gaussian it is the covariance matrix) will be computed. Again, several choices exist, such as Parzen windows [20]. The approach enforces the same circular kernel for all of the data points. Thus, there is only one parameter σ (*i.e.* $\Sigma = \sigma I$), which is set such that the mean log likelihood of every point is maximized using a leave-one-out scheme.

It is important to note that the statistical consistency of non-parametric density estimation holds for any choice of a kernel (*e.g.* Gaussian, Rectangular, Epanechnikov), even if it is restricted to being circular. In the following subsection we discuss the consistency of marginals and derive a method for estimating the kernel parameters.

4.2 Consistency of Marginals

It is essential that the one- and two-node marginals that form the tree distribution be consistent, otherwise the conditional distributions would not be true densities. Given that we enforce circular kernels (*i.e.* diagonal covariance matrices) in our density estimates, consistency in this case implies that the variance along a dimension in a two-node marginal should be equal to the variance of its one-node marginal. For example, consider the model depicted in Figure 2(b). The joint distribution of all three features under this model is

$$P(X_A, X_B, X_C) = \frac{P(X_A, X_B)P(X_B, X_C)}{P(X_B)}. \tag{8}$$

Thus enforcing the consistency condition requires that the variance of X_B under both joints and the marginal be approximately the same. Note that issues regarding inconsistencies only arise when we do not have enough data points.

To enforce consistency, we set the diagonal covariance matrix of the two-node marginals to the variances of the component one-node marginals. Thus, the covariance matrix for an arbitrary two-node marginal $P(X,Y)$ is

$$\Sigma_{xy} = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}, \tag{9}$$

where σ_x^2 is the variance of the one-node marginal $P(X)$ and σ_y^2 is the variance of the one-node marginal $P(Y)$. This is justified by the fact that marginalizing a bivariate Gaussian with a covariance matrix of the above form results in two univariate Gaussians (one with variance σ_x^2 and another with variance σ_y^2) which implies consistency.

Now we need to optimize the n variance parameters for the tree. Ideally, we should select the n parameters such that the mean log likelihood of every point (under the tree distribution) is maximized using a leave-one-out scheme. However, for high-dimensional feature spaces this method becomes computationally expensive. Alternatively, we maximize each of the variance parameters individually using the Parzen windows technique and then we introduce a single multiplicative constant k to add another degree of flexibility to our model. This constant allows us to change the variance values for all of the one-node marginals by a constant factor. We optimize k such that the mean log-likelihood of the data points under the model is maximized using a leave-one-out scheme. This allows us to use the structure of the tree to influence the variances while avoiding a complex optimization procedure.

4.3 Estimating Mutual Information

When computing the cost matrix, we need to compute the mutual information between random variables. The mutual information between two random variables X and Y is defined as

$$I(X;Y) = h(X) + h(Y) - h(X,Y), \tag{10}$$

where $h(X)$ and $h(Y)$ are the differential entropy [7] of the marginal distributions and $h(X,Y)$ is the differential entropy of the random vector (X,Y). Since we only have samples from a distribution (*i.e.* its exact form is unknown) we cannot solve for exact marginal and joint entropies. One could resort to assuming a specific parametric form or discretizing the data, but such procedures either misrepresent the data or throw away a lot of information. Thus, we use the resubstitution estimate of entropy [3], which does not make any assumptions about the data. Under this estimate, the entropy (in nats) of a distribution given k samples from that distribution is

$$\hat{h}_k = -\frac{1}{k}\sum_{i=1}^{k} ln\, f_k(x_k), \tag{11}$$

where f_k is a non-parametric density estimate of the k sample points.

5 Synthetic Data

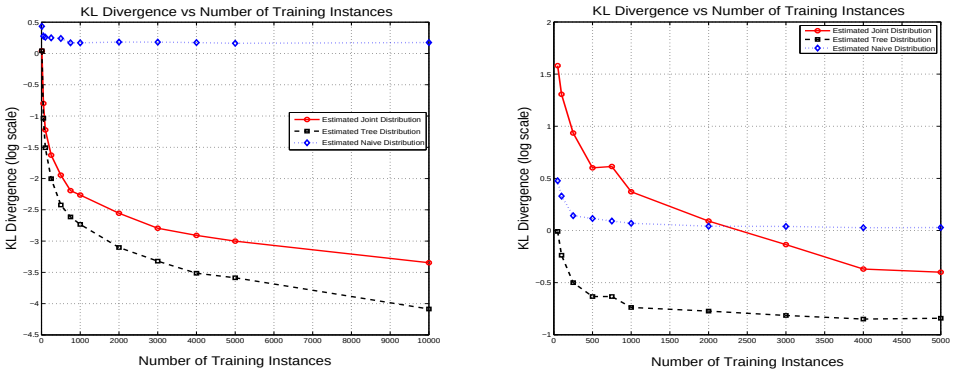
In this section we support the claims that a tree model will always better approximate the joint density than the Naive Bayes model. We also test that the tree model converges to the best possible estimate of the true distribution much faster than estimating the joint distribution directly.

We generated samples from a known multivariate distribution. In the first test, we chose a simple distribution where the exact conditional independence can be represented by a tree model. In the second test, we chose a more complex distribution such that the tree model can provide a good (not exact) approximation to the true distribution. We varied the number of training instances and at each run computed three estimates of the joint distributions: the Naive Bayes estimate (using non-parametric marginals), the tree estimate, and the direct non-parametric estimate. We then measured the KL divergence from the true distribution to each of the three estimates for varying training set sizes. Let $D(P||P_e)$ denote the KL divergence from the true distribution, P, to an estimate of the distribution, P_e. Recall that the KL distance is defined as

$$D(P||P_e) = E_p\left(log\frac{P(X)}{P_e(X)}\right). \tag{12}$$

Given a large number of samples from the true distribution (which we can easily generate), then by the law of large numbers we can estimate the KL divergence by the average log of the likelihood ratios. That is, given n samples (where n is large) the KL divergence can be estimated as

$$D(P||P_e) = \frac{1}{n}\sum_{i=1}^{n} log\frac{P(x_i)}{P_e(x_i)}. \tag{13}$$



(a) (b)

Figure 3: Convergence plots of various density estimates of a 3-dimensional (left) and 10-dimensional (right) Gaussian.

The plots of KL divergence versus number of samples for both experiments are shown in Figures 3(a) and 3(b). We analyze these plots in more detail in the following subsections.

5.1 Experiment 1: Simple Case

We generated samples from a 3-dimensional Gaussian whose exact conditional independence relations are shown in Figure 2(b). The mean of the Gaussian was placed at the origin. Recall that zeros in the inverse covariance matrix correspond to missing edges in a graphical model and thus control the conditional independence relations among the random variables [15, 4]. Thus, to model the conditional independence relationships in Figure 2(b), entries $(1,3)$ and $(3,1)$ of the inverse covariance matrix were set to zero and the other entries were chosen arbitrarily such that the covariance matrix remains positive definite.

Figure 3(a) shows the KL divergence of the true distribution to the three estimates. The following points summarize our analysis of the graph.

- As expected the Naive model converges to a distribution that is far away from the true distribution. This is because dependencies exist in the true distribution which are not modeled by the Naive model.
- Due to statistical consistency of non-parametric estimates, the direct estimate of the joint will converge to the true distribution (*i.e.* KL = 0). Since our tree distribution models the exact conditional independence relations, it will converge faster than the direct estimate of the joint (as shown in the plot) because we are estimating lower dimensional marginals.
- The KL divergence of the tree distribution is always smaller than the Naive model, supporting our claim that using a tree model will always perform better than a Naive Bayes model. This result can be extended to parametric models. If the assumed parametric form is not consistent with the true distribution, then in the limit the estimated parametric distribution will converge to a distribution that is far away (*i.e.* larger KL divergence) from the true distribution.

Given that the joint distribution is only one dimension greater than the two-node marginals, we would not expect a significantly faster convergence rate for the tree distribution. In the next experiment, in which the joint distribution is in 10 dimensions, the advantages of using the tree distribution will become even more more apparent.

5.2 Experiment 2: Complex Case

For this case we generated samples from a 10-dimensional Gaussian, whose approximate conditional independence relationships are shown in Figure 2(c). The following points summarize our analysis.

- With a small number of training instances the Naive model had a closer approximation to the true distribution than the direct joint estimate. This is because density estimates in high dimensions with few points usually leads to inaccurate estimates.

(P) Category Name	# of images	(Z) Category Name	# of images
Pennate diatoms	124	*Calanus finmarchicus*	132
Ciliates	179	*Conchoecia* Ostracods	100
Non-cell	113	Euphausiids	131
Mesodinium	71	Pteropods	142
Skeletonema	169	Larvaceans	133
Thallasiosira	86	Small Copepods	433
Pseudo-nitzschia	61	Unidentified Cladocerans	108
		Siphonophores	202

Table 1: Taxonomic categories for the 7-class Phytoplankton data set (columns 1 and 2) and the 8-class Zooplankton data set (columns 3 and 4).

- Again, the tree distribution converges rapidly to the best approximation of the true distribution. The KL divergence remains approximately constant after 1000 sample points, while the direct estimate of the joint was not as accurate with 5000 sample points.
- Again, the tree estimate was always better than the Naive estimate and the Naive estimate converged to a distribution that is far away from the true one.

These plots highlight the main advantages of non-parametric tree-structured distributions. In the following section we compare the performance of the proposed tree-model versus other parametric and non-parametric estimates on a real-world plankton classification data set. We also discuss the bias/variance trade off we incur by moving from the Naive Bayes model to this tree-structured distribution.

6 Plankton Classification

We are provided with a training set of images and their corresponding class labels and the goal is given a new unseen image to assign a class label to that image. The assumption is that each image contains only one plankton organism.

We tested five models, a maximum likelihood (ML) Gaussian estimate of the full joint distribution (Gaussian - Joint), a Naive Bayes model with ML Gaussian estimates for the marginals (Gaussian - Naive), a Naive Bayes model with non-parametric marginals (NPD - Naive), a direct non-parametric estimate of the joint distribution (NPD - Joint) and the tree estimate with non-parametric marginals (NPD - Tree).

6.1 Data Sets and Features

We tested all five models on two different data sets (Table 1). The first one is composed of 7 phytoplankton categories [5] and the second is composed of 8 zooplankton categories [16]. We have found that a simple global bimodal segmentation is usually effective for separating the plankton from the background, which tends to be significantly darker than the object. We use expectation maximization (EM) to fit a mixture of two Gaussians to the histogram of gray values for a given image [8]. The Bayesian decision boundary defines the cut point between foreground and background. After that, morphological hole filling [21] is used to capture the stray dark pixels inside the object.

We computed the following 20 features for the phytoplankton data set:

- *Simple Shape:* The 8 features in this category include, Area, Perimeter, Compactness, Eigenratio, Eccentricity, Standard deviation of area across connected components, Convexity, and Rectangularity.
- *Moments of Intensity Histogram:* The 5 features here include, the mean, std, skewness, and kurtosis of the histogram of grayscale values and the entropy of the normalized histogram.
- *Moment Invariants:* These 7 features are basically the first seven moment invariants as proposed by Hu (actually they are the log of the moments). The moments are computed over the binary image so they are shape descriptors.

For the zooplankton data set we computed the Shape Index texture features [18]. A function of the image called the shape index is computed: $S(p,\sigma) = \arctan[\frac{\kappa+\mu}{\kappa-\mu}](p,\sigma)$, where κ is the isophote curvature of the intensity surface, and μ is the flowline curvature. The curvatures are computed via combinations of image derivatives,

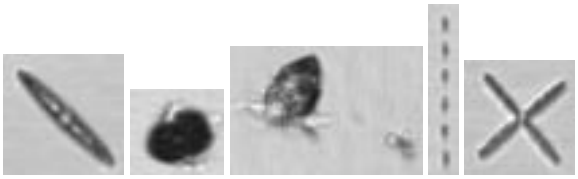


Figure 4: Sample phytoplankton images

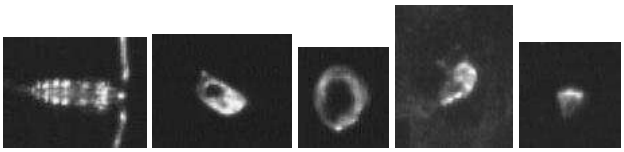


Figure 5: Sample zooplankton images

which are computed using the Gaussian derivative filters. We calculate the shape index at every pixel in the image at a range of scales, and aggregate the values into a histogram by quantizing the shape index. Before we do this, however, we ignore areas of low curvature by excluding points where the isophote (flowline) is below the mean isophote (flowline). Currently histograms are calculated with $\sigma = \{\sqrt{2}, 2, 2\sqrt{2}\}$ and 40 bins, yielding 120 features.

6.2 Experimental Results

We performed 10-fold stratified cross validation to test each of the five models. For each data set we used the exact same folds for all of the models we tested on. The results are summarized in Table 2. Our analysis of the results are summarized in the following points.

- The tree-structured non-parametric estimate performed on average better than the other four models it was tested on. In the case of the zooplankton data set it performed significantly better (11% better than the next best result). Part of the increased performance on the zooplankton data set is due to the fact that it contains more images per class.
- When comparing all the non-parametric models, the accuracy and standard deviation numbers reported agree with our intuition regarding the bias/variance trade-off. The joint model (NPD-Joint) has the highest variance out of all three non-parametric models. Also, the naive model (NPD - Naive) has the lowest variance. Thus, the tree-based model provide a much lower bias estimate, at the cost of a small increase in variance from the naive model. For, example in the case of the zooplankton data set, the accuracy of the tree-based estimate was 11% higher than the naive model, while its standard deviation only increased by 0.35%.

Overall, these results support our claim that there is a place for such non-parametric models in the vision community and that one can obtain a much better estimate of the underlying distribution by using a tree-structure.

Method	Accuracy	STD	Accuracy	STD
Gaussian - Joint	69.13%	6.81%	30.33%	2.60%
Gaussian - Naive	56.46%	5.87%	30.40%	2.59%
NPD - Joint	49.81%	7.67%	32.23%	9.20%
NPD - Naive	66.87%	6.20%	32.73%	2.31%
NPD - Tree	**70.26%**	6.37%	**44.42%**	2.67%

Table 2: Results on the two plankton data sets. The second and third columns are the results pertaining to the 7-class phytoplankon data set, while the last two columns are the results pertaining to the 8-class zooplankton data set.

7 Conclusions

In conclusion, we proposed a simple MAP classification procedure that models part of the dependencies between the features. We argued that such a model overcomes the main issue with non-parametric estimation regarding having enough samples. We analyzed the performance of this model under synthetic distributions and real-world data sets and in all of the cases we tested on, the results came out to favor the model that we proposed. We hope that these results will influence the choice of modelers in the future to consider using such powerful non-parametric models.

Acknowledgments

We thank Bigelow Laboratory for Ocean Sciences and Professor Mark Benfield for providing the plankton images. Special thanks to current and previous members of the Plankton Classification project for their contributions. This research was supported by the National Science Foundation under grant ATM-0325167.

References

[1] S. Altmueller and R. Haralick. Approximating high dimensional probability distributions. In *International Conference on Pattern Recognition*, 2004.

[2] F. Bach and M. Jordan. Beyond independent components: Trees and clusters. *Journal of Machine Learning Research*, pages 1205–1233, 2003.

[3] J. Beirlant, E. Dudewicz, L. Gyorfi, and E. van der Meulen. Nonparametric entropy estimation: An overview. *International Journal of the Mathematical Statistics Sciences*, 6:17–39, 2001.

[4] Jeff Bilmes. Factored sparse inverse covariance matrices. In *International Conference on Acoustices, Speech, and Signal Processing*, 2000.

[5] M. Blaschko, G. Holness, M. Mattar, D. Lisin, P. Utgoff, A. Hanson, H. Schultz, E. Riseman, M. Sieracki, W. Balch, and B. Tupper. Automatic in situ identification of plankton. In *Proceedings of the IEEE Workshop on Applications of Computer Vision*, 2005.

[6] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

[7] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley and Sons, 1991.

[8] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–38, 1977.

[9] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, Inc, 2001.

[10] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29, 1997.

[11] D. Heckerman. *Probabilistic Similarity Networks*. MIT Press, 1991.

[12] D. Heckerman. A tutorial on learning bayesian networks. Technical report, Microsoft Research Technical Report MS-TR-95-06, 1995.

[13] H.P. Jeffries, M.S.Berman, A.D Poularikas, C. Katsinis, I. Melas, K. Sherman, and L. Bivins. Multisource data classification with dependence trees. *IEEE transactions on geoscience and remote sensing*, 40:609–617, 2002.

[14] S. Kirshner, P. Smyth, and A. Robertson. Conditional chow-liu tree structures for modeling discrete-valued vector time series. In *Uncertainty in AI*, 2004.

[15] S. Lauritzen. *Graphical Models*. Oxford Science Publications, 1996.

[16] Dimitri A. Lisin, Marwan A. Mattar, Matthew B. Blaschko, Mark C. Benfield, and Erik G. Learned-Miller. Combining local and global image features for object class recognition. In *Proceedings of the IEEE CVPR Workshop on Learning in Computer Vision and Pattern Recognition*, 2005.

[17] F. Pernkopf and J. Bilmes. Discriminative versus generative parameter and structure learning of bayesian network classifiers. In *Intl. Conf. on Machine Learning*, 2005.

[18] S. Ravela. *On Multi-Scale Differential Features and their Representations for Image Retrieval and Recognition*. PhD thesis, University of Massachusetts Amherst, 2002.

[19] B. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[20] B. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall: London, 1986.

[21] P. Soille. *Morphological Image Analysis: Principles and Applications*. Springer-Verlag, 1999.