Detection of Man-made Structures in Natural Images

Tim Rees

December 17, 2004

Abstract

Object detection in images is a very active research topic in many disciplines. Probabilistic methods have been applied to the problem with varying degrees of success. A logistic classifier, Markov random field (MRF), and discriminative Random Field (DRF) were used for the detection of man-made structures in natural images. It was found that the MRF and DRF models were often improvements over the logistic model, but they introduced some false-positives. The Matlab framework developed during the course of the investigation should serve as an excellent staging ground for future explorations of many sub-problems.

1 Introduction

Object detection in images a very active area of research in fields such as computer vision, pattern recognition, machine learning, and image processing. While many schemes exist, using Markov random fields (MRFs) has been growing in popularity and effectiveness [5]. Variations on Markov random fields are also being investigated, discriminative (or conditional) random field (DRFs)approaches have recently been attempted with claims of success. This report will present an attempt to reproduce the results of Kumar and in the detection of man-made structures in natural images using MRFs and DRFs [2], [4], and [1].

Image classification is an interesting study because it combines so many challenging disciplines. Image processing techniques are required to extract features from images, probabilistic models are constructed to model spatial dependencies, machine learning and optimization techniques are required for learning parameters of models, and approximate inference strategies must be employed to utilize the final result on test images.

A Matlab framework was developed to accomplish the task of man-made structure detection in natural images. Specific details of the implementation are not included here, but the source code is fully available from *www.cs.ubc.ca/ trees*. This document will focus on the methodology behind the development, how it might be improved, and some of the results that were obtained. Each of the major tasks involved will be discussed in detail.

2 Image Models

The description of an image will follow the notation and work of [1], Images are composed of *sites* (not necessarily individual pixels), and the classification of an image consists of determining the correct labels of each site in an image. Letting x_i denote the label of the i^{th} image site, then $x_i \in \{-1, 1\}$, indicating a site is natural or man-made, respectively. Observed data from an image site i is represented by y_i, and observed data is generated from the feature vectors of the image sites.

Before proceeding to the model of an image the following definition for a DRF (or CRF) is given (taken directly from [1]).

Definition of a CRF/DRF *Let G=(S,E) be a graph such that x is indexed by the vertices of G. Then (x,y) is said to be a conditional random field if when conditioned on y, the random variables* x_i *obey the Markov property with respect to the graph:* $p(x_i|y, x_{S-\{i\}}) = p(x_i|y, x_{N_i})$*, where* $S - \{i\}$ *is the set of all nodes in G except the node i,* N_i *is the set of neighbours of the node i in G, and* x_Ω *represents the set of labels of the nodes in set* Ω.

When modeling an image with a DRF, the vertex set corresponds to the set of image sites, and the edge set corresponds to the connections between neighbouring sites. In their DRF model for images, Kumar and Hebert use the Hammersley-Clifford theorem and the assumption that only pairwise clique potentials are non-zero, that is, only immediate neighbours interact. From this they obtain a joint distribution over the labels given observations y defined by:

$$p(x|y) = \frac{1}{Z} exp(\sum_{i \in S} A(x_i, y) + \sum_{i \in S} \sum_{j \in N_i} I(x_i, x_j, y))) \tag{1}$$

Here Z is a normalizing factor referred to as the partition function. Kumar and Hebert refer to $A(x_i, y)$ and $I(x_i, x_j, y)$ as the *association* and *interaction* potentials, respectively. $A(x_i, y) = log\sigma(x_i w^T h_i(y))$, where $\sigma(\alpha) = \frac{1}{1+e^{-\alpha}}$ is chosen as the association potential. This is a somewhat arbitrary choice, and future investigations should consider alternatives.

The logistic classifier model of an image does not incorporate any interaction between neighbouring image sites, therefore $I(x_i, x_j, y)$ is identically 0. This simplification results in an expression for the local log-likelihood of an individual image site: $p(x_i|y) = log(\sigma(x_i w^T h_i(y)))$. The only parameters of the model are the elements of the vector w. The length of w must be the same as the transformed observed feature data (see section 3), which has a lead element of 1 to allow for a constant w_0 term.

The Ising MRF model is a simple extension of the logistic model that has $I(x_i, x_j, y) = vx_ix_j$, for some scalar parameter v. The parameters w and v must be determined for the model based on a training data set. An Ising MRF model with $v = 0$ reduces to the logistic model.

The full DRF model is an extension of the Ising MRF. Instead of constant edge potentials between neighbours, the DRF computes $I(x_i, x_j, y) = x_ix_jv^T\mu_{ij}(y)$, with the first element of $\mu_{ij}(y) = 1$ Note that the edge potentials are *not* symmetric. The vector $\mu_{ij}(y)$ is defined in section 3, but note that if v's elements are indexed from 0 to $n-1$, and $v_1...v_{n-1} = 0$, the DRF model degenerates to an Ising MRF. For all three models the image sites consisted of 16×16 pixel blocks.

3 Feature Extraction

The features used were a combination of single-scale and multi-scale features (which are defined below), computed from orientation histograms of the gradient. Before computing the different features the magnitude and orientation of the gradient of the input image was taken. Each image was segmented in 16×16 pixel blocks (the image sites), and for each site single-scale and multi-scale features were calculated.

3.1 Single-Scale Features

The orientation and magnitude of the image gradient at each image site was used for feature extraction. A weighted histogram of the gradient orientation was computed at each image site. The orientation of the gradient of each pixel in a block was binned with a weight equal to the magnitude of the gradient at that pixel. Once the weighted histogram was computed, kernel smoothing was applied. With N

being the total number of bins in the histogram, n_i the count of the i^{th} bin, and a symmetric positive kernel smoothing function $K(x)$ with bandwidth bw, the smoothed bin counts were given by:

$$n'_j = \frac{\sum_{i=1}^{N} K((n_j - i)/bw) n_i}{\sum_{i=1}^{N} K((n_j - i)/bw)}. \tag{2}$$

$K(x) = e^{-x^2}$ was used, but it would be an interesting investigation to see the impact of the smoothing function on the classifier performances. The number of bins used was set to 50, another exploration might involve finding the optimal number of bins to use.

From the smoothed bin counts the single-site features were extracted. The first three features were heaved central shift moments where the p^{th} moment is given by:

$$v_p = \frac{\sum_{j=1}^{N} (n'_j - v_0)^{p+1} H(n'_j - v_0)}{\sum_{j=1}^{N} (n'_j - v_0) H(n'_j - v_0)} \tag{3}$$

Here v_0 was taken to be the mean histogram magnitude, $v_0 = \frac{1}{N}\sum_{i=1}^{N} n'_i$, and the Heavyside function was used for H. The first three features at the single-scale were formed by v_0, v_1, and v_2.

Two other orientation-based features were computed at the single-scale. The first orientation feature used was the relative location of the peak of the histogram (in radians). The second orientation based feature was give by $|sin(p_1 - p_2)|$, where p_1 is the relative location of the peak of the orientation histogram, and p_2 is the location of the second-highest peak of the histogram. Passing the difference of the angles through a sinusoid should favour the presence of right angles (since $sin(\pi/2) = 1$.

3.2 Multi-Scale Features

The multi-scale features computed at each site were a simple extension of the single-scale features. To get a measure of interaction between sites and their neighbours, multi-scale features were extracted from orientation histograms of block sizes 16×16, 32×32, and 64×64, about the center of each image site (of course edge sites were exceptions). At each scale, the same features were calculated as in the single-scale case. This would lead to a total of 15 multi-scale features, but instead of using the mean magnitude of the histogram in each scale, the *difference* in magnitude was taken from the first scale to the second and third, reducing the number of features at the multi-scale by one. A total of 14 mulit-scale features were computed at each image site. It is important to note that the windows in the multiple scales do not perfectly align with the divisions between sites.

3.3 Computing $h_i(y)$ and $\mu_{ij}(y)$

As seen in the models of section 2, the feature data was not directly used. Instead, for both the association and interaction potentials *transformed* feature vectors were used. To compute the transformed feature vector $h_i(y)$, the following was used:

$$h_i(y) = [1, y, f(y)]^T \tag{4}$$

Where $f(y)$ is the vector consisting of all cross-combinations of the elements of y. That is, if there were n elements in y then $f(y)$ would have $\frac{n(n+!)}{2}$ elements $y_i y_j$.

The vector μ_{ij} was computed by simply taking the absolute value of the difference between the expanded multi-scale features of sites i and j, and then resetting the lead element back to 1. Kumar and Hebert suggest the concatenation of the expanded multi-scale feature vectors of sites i and j, but this doubles the number of parameters that must be learned for the DRF model, so it was not attempted.

3.4 Interpretation

Some of the features used are intuitively reasonable, for instance the orientation features. The moment based features are not necessarily obvious choices for features in the search for man-made structures. Does the presence of high magnitude gradients in an image site indicate a building? Certainly it might result from a lot of edges, butt edges exist in nature too. The developed platform should be used as a testing ground for feature selection in object specific detection.

The reasoning behind the expansion in of the feature vectors is not entirely clear, other than to add some flexibility to the models.

4 Parameter Learning

In section 2 three models were given for the binary classification of images. Each model featured parameters v and w, which were learned from M training images ($M = 108$ was used).

In the case of the logistic classifier, no interactions between neighbouring sites existed, so v was known to be zero. Learning the parameters w was fairly simple, the objective to maximize was convex and simple enough to differentiate by hand, so the Hessian could be constructed and Newton's method was applied. The log classifier parameters were learned by maximizing the objective:

$$L(w) = \sum_{m=1}^{M} \sum_{i \in S} log\sigma(x_i(m)w^T h_i(y(m)))(5)$$

With Hessian (given in [4]):

$$\nabla^2 L(w) = -\sum_{m=1}^{M} \sum_{i \in S} \{\sigma(w^T h_i(y))(1 - \sigma(w^T h_i(y)))x_i h_i(y)\}. \qquad (6)$$

The logistic parameters computed for w served as an initial guess for the optimal w in each of the MRF and DRF models. The optimal parameters for the MRF and DRF models were found by optimizing:

$$\theta_{max} = arg \max_{\theta} \sum_{m=1}^{M} \sum_{i \in S} \{log(\sigma(x_i w^T h_i(y))) + \sum_{j \in N_i} x_i x_j v^T \mu_{ij}(y) - log(z_i)\}. \qquad (7)$$

With:

$$z_i = \sum_{x_i \in \{-1,1\}} exp(log(\sigma(x_i w^T h_i(y))) + \sum_{j \in N_i} x_i x_j v^T \mu_{ij}(y)). \qquad (8)$$

Optimizing this objective function led to the values of the parameters that give the maximum likelihood for the known label configurations of the training data. In the case of the MRF model, v was just a scalar and $\mu_{ij} = 1$. To compute v, a gradient ascent method was used. Such methods are much slower than Newton's method, but when the Hessian is unavailable they are necessary. Tests were conducted with varying guesses for the initial value of v, and for the training data v always converged to 0.5598.

Maximizing equation (7) for the DRF model was much slower because for the DRF v had a length of 120, matching the length of μ_{ij}. This significantly increased the solving time for the DRF parameters. Alternative optimization techniques should be considered.

5 Inference

With model parameters learned, test image classification becomes possible. In the case of the logistic classifier, image segmentation was a relatively easy task. The log-classifier does not take into account any interactions between sites, so the optimal label configuration for an image was just the configuration where each site had been assigned its most likely label.

Determining the most likely label configuration for models with interactions between sites is an NP-hard problem. To label an image with exact inference, the likelihood of all possible label configurations would have to be computed, and from that the best configuration could be chosen. Even with 256×384 images segmented into 16×16 blocks this would required computing the likelihood of $2^{16*24} = 2^{384}$ configurations! To counter this problem, approximate inference techniques were used.

The approximate inference portion of the project was done in collaboration with Sohrab Shah and Frank Hutter. It was a relatively simple extension to make the projects compatible, and the work was mutually beneficial. Various approximate schemes were used, loopy belief propagation, generalized loopy belief propagation, Gibbs sampling and other methods were tried. Generalized loopy belief propagation generally tended to give good results and was used for both the MRF and DRF. For specific details on the mechanics of the approximate inference work, please refer to the report by Hutter and Shah.

6 Results

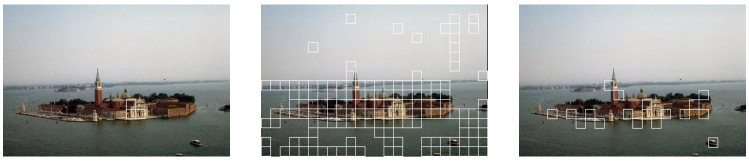

Figure 1: From left to right, the logistic, MRF, and DRF classification of a sample image.

In some situations, the log classifier, MRF, and DRF models performed quite well. In general, though, it was seen that the log classifier, while introducing very few false-positives, did not capture the majority of the man-made structures in the images. The MRF model on the other hand, often introduced a lot of false-positives, and seemed to be too liberal in propagating similarity between neighbours. The DRF model lay somewhere between the two, in some situations it did an amazing job of capturing the man-made structures and identifying very few false-positives. Figure 1 illustrate these observations. The DRF model did not always beat the MRF, as is seen in figure 2. The tendency of neighbouurs to be similar in the MRF model was sometimes advantageous.

The presence of trees, tree-branches, and horizon lines tended to spoil the performance of all classifiers, especially the MRF. This problem seems to indicate that the features chosen to distinguish man-made structures were not selective enough. It is recommended that the feature choices be revised.

Rooves were rarely detected by any of the three models. Generally a roof blends in with the background more than the body of a structure, and Kumar and Hebert note that roof-top detection is unlikely [2]. Introduction of additional feature to identify roof-tops should be considered.

A full set of test results for each model along with the original images can be found at http://www.cs.ubc.ca/ trees under the Cs532C link.

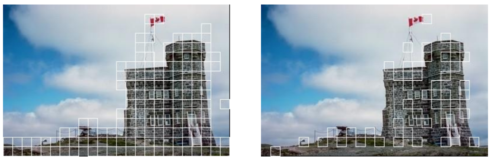

Figure 2: An example of the MRF (left) outperforming the DRF (right).

7 Future Work

In some cases the success of the MRF and DRF models was impressive, but the developments should only be considered a first step. A strong foundation is now available to use in explorations of many of the major disciplines within graphical models and machine learning research. The Matlab implementation was made to be modular, so individual processes can be interchanged and their impact studied immediately.

The feature extraction process should be re-evaluated and refined. At present the features have been designed to help detect man-made structures, but theoretically any object should be detectable if the right features are extracted. An exploration of the methods in detecting other objects should be conducted. The features used for man-made structure detection could also be improved, colour features could be added, and a study of features that distinguish man-made structures from other structured objects (such as trees, horizon lines) should be conducted.

The current framework was also used as a testing ground for approximate inference techniques. The image classification problem could help in the empirical analysis of the approximate inference routines.

Image model design and parameter learning could also be tested on the current framework. In [3] approximate parameter learning methods are described, it might be worthwhile studying the impacts of an approximate learning approach.

8 Conclusion

A strong foundation for binary classification of natural images was constructed in Matlab. A log classifier, Markov random field, and discriminative random field were investigated for the task of detecting man-made structures in images. Generally, the DRF provided the best results followed by the MRF, but both techniques could be improved. There Matlab framework that was developed should serve as an excellent platform for testing the effectiveness of feature selection, model design, parameter learning, and approximate inference techniques.

References

[1] Sanjiv Kumar and Martial Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. IEEE International Conference on Computer Vision, 2003.

[2] Sanjiv Kumar and Martial Hebert. Man-made structure detection in natural images using a causal multiscale random field. Utah, 2003. IEEE International Conference on Computer Vision and Pattern Recognition.

[3] Sanjiv Kumar and Martial Hebert. Approximate parameter learning in discriminative fields. Utah, 2004. Snowbird Learning Workshop.

[4] Sanjiv Kumar and Martial Hebert. Discriminative random fields for modeling spatial dependencies in natural images. Advances in Neural Information Processing Systems, NIPS 16, 2004.

[5] Patrick Perez. Markov random fields and images. CW Quarterly, Volume 11, 1998.