CRFs for Image Classification

Devi Parikh and Dhruv Batra
Carnegie Mellon University
Pittsburgh, PA 15213
{dparikh,dbatra}@ece.cmu.edu

Abstract

We use Conditional Random Fields (CRFs) to classify regions in an image. CRFs provide a discriminative framework to incorporate spatial dependencies in an image, which is more appropriate for classification tasks as opposed to a generative framework. In this paper we apply CRFs to two image classification tasks: a binary classification problem (man-made vs. natural regions in the Corel dataset), and a multiclass problem (grass, sky, tree, cow and building in the Microsoft Research, Cambridge dataset). Parameter learning is performed using Mean Field (MF) and Loopy Belief Propagation (LBP) to maximize an approximation to the conditional likelihood, and inference is done using LBP. We focus on three aspects of the classification task: feature extraction, feature aggregation, and techniques to combine binary classifiers to obtain multiclass classification. We present classification results on sample images from both datasets and provide analysis of the effects of various design choices on classification performance.

1 Introduction

Standard approaches to texture based image region classification often require a homogeneous region of interest to be identified in the image to be classified [1]. To avoid this human intervention, a test image is first segmented into homogeneous segments and then each segment is individually classified. This is often prone to segmentation errors. The other approach is to over-segment the image into superpixels [2], and classify each superpixel based on it's texture. However, natural images exhibit spatial smoothness, and hence this should be exploited rather than classifying each of the superpixels individually. Markov Random Fields (MRFs) allow the consideration of these spatial dependencies in a principled manner, and hence have been used extensively for various region classification applications in computer vision [3]. MRFs, however, are generally used in the generative probabilistic framework where joint probability distribution of the observed data and the corresponding labels is modeled. The posterior over the labels given the data is expressed using the Bayes' rule, where the prior over labels is modeled as a MRF and for computational tractability, the likelihood model is assumed to be a fully factorized form.

However, this assumption is too restrictive for several applications in computer vision. This is because natural images exhibit spatial order and thus not only is an observation at a particular site highly correlated with the observations at the surrounding sites, these observations typically do not become independent even conditioned on the labels. While

it is important for the model to allow for tractable inference, it is undesirable to make unwarranted assumptions. As stated in [4], one way to satisfy both requirements is to model the conditional distribution over the labels given the observation data, instead of the joint probability distribution over both labels and the observations. This is the discriminative framework employed by Conditional Random Fields (CRFs) [5]. Arbitrary attributes of the observations can be captured via this model by avoiding the factorization assumption made for tractable inference on MRFs. In addition, for a classification task, the goal is to assign a label to a novel set of observations (image) that maximizes the conditional probability of the labels given the observations. This posterior distribution is often simple to model, while the underlying joint distribution may be quite complex.

Kumar *et al*. [6] propose an enhancement to CRFs, called Discriminative Random Fields (DRFs), and use local discriminative models to capture the class associations at the individual sites as well as the interactions with the neighboring sites.

The rest of the paper is organized as follows. Section 2 provides background information about CRFs and DRFs. Section 3 discusses our overall approach, Section 4 presents the experimental results and Section 5 concludes the paper.

2 Conditional Random Fields: Background and Notation

Let $\boldsymbol{x}$ be the random vector over the observed data, the components of which, x_i describe the data at site i, $x_i \in \Re^c$. Let $\boldsymbol{y}$ be the random vector over the label sequences, where every component $y_i \in \boldsymbol{\mathcal{Y}}$. We deal with binary classifiers, and in our case $\boldsymbol{\mathcal{Y}} \doteq \{ \ -1 \quad 1 \ \}$. The definition of Conditional Random Fields, as proposed by Lafferty *et al*. [5], is discribed below:

CRF Definition: *Let $G = (V, E)$ be a graph such that $\boldsymbol{y} = (y_v)_{v \in V}$, so that $\boldsymbol{y}$ is indexed by the vertices of G. Then $(\boldsymbol{x}, \boldsymbol{y})$ is a conditional random field if, when conditioned on $\boldsymbol{x}$, the random variables y_v obey the Markov property with respect to the graph:*

$$p(y_v \mid \boldsymbol{x}, y_{V-\{v\}}) = p(y_v \mid \boldsymbol{x}, y_{\mathcal{N}_v}), \tag{1}$$

where $V - \{v\}$ is the set of all nodes in G other than node v, and $\mathcal{N}_v$ is the set of neighbors of the node v in G.

A CRF is thus a random field globally conditioned on the observation vector $\boldsymbol{x}$. Using the Hammersley Clifford theorem [7] and assuming only upto pairwise clique potentials to be non-zero, the joint distribution over the labels $\boldsymbol{y}$ given the observations $\boldsymbol{x}$ can be written as:

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{1}{Z} exp \left(\sum_{i \in V} A_i(y_i, \boldsymbol{x}) + \sum_{i \in V} \sum_{j \in \mathcal{N}_i} I_{ij}(y_i, y_j, \boldsymbol{x}) \right), \tag{2}$$

where Z is the normalizing constant known as the partition function, and A_i and I_{ij} are the negative unary (association) and pairwise (interaction) potentials respectively.

In the CRF framework Lafferty *et al*. [5] propose and use a fixed feature function as the association potential. These could be, for instance, real-valued functions taking on the value of a feature for a particular range of values and zero otherwise. In the DRF framework Kumar *et al*. [6] model the association potential as a posterior probability of the class labels given the observation, with the parametric form:

$$A_i(y_i, \boldsymbol{x}) = log(\sigma(y_i \boldsymbol{w^T} \boldsymbol{h_i}(\boldsymbol{x}))), \tag{3}$$

where,

$$\sigma(net) = \frac{1}{1 + e^{-(net)}}, \tag{4}$$

and

$$\boldsymbol{h}_i(\boldsymbol{x}) = [\, 1,\ \phi_1(\boldsymbol{f}_i(\boldsymbol{x})), \ldots,\ \phi_R(\boldsymbol{f}_i(\boldsymbol{x}))\,], \tag{5}$$

where $\phi_k(.)$ are arbitrary non-linear transforms of the feature functions $\boldsymbol{f}_i(.)$, which themselves map the observations $\boldsymbol{x}$ onto a feature vector, so that $\boldsymbol{f}_i\ : \boldsymbol{x} \mapsto \Re^l$, and $\boldsymbol{w}$ are the parameters to be learnt. We use filter-bank responses (explained in detail in Section 3) as our feature functions $\boldsymbol{f}_i(.)$, and a quadratic kernel as the non-linear transform $\phi_k(.)$.

In the CRF framework, interaction potentials are also chosen as fixed real-valued feature functions. As opposed to that, in the DRF framework, the interaction potential is represented as a pairwise discriminative model of the form:

$$I_{ij}(y_i, y_j, \boldsymbol{x}) = y_i y_j \boldsymbol{v}^{\boldsymbol{T}} \boldsymbol{\mu}_{ij}(\boldsymbol{x}), \tag{6}$$

where, for a pair of sites (i, j), $\boldsymbol{\mu}_{ij}(\boldsymbol{\psi}_i(\boldsymbol{x}), \boldsymbol{\psi}_j(\boldsymbol{x}))$, (called $\boldsymbol{\mu}_{ij}(\boldsymbol{x})$, for simplicity) is the pairwise feature vector, $\boldsymbol{\psi}_i(\boldsymbol{x})$ is another feature function at site i, and $\boldsymbol{v}$ are the parameters to be learnt. In our case, we use $\boldsymbol{\psi}_i(\boldsymbol{x})$ to be $\boldsymbol{h}_i(\boldsymbol{x})$, and concatenate $\boldsymbol{\psi}_i(\boldsymbol{x})$, and $\boldsymbol{\psi}_j(\boldsymbol{x})$, to form the pairwise feature vector $\boldsymbol{\mu}_{ij}(\boldsymbol{x})$ at the edge.

Since exact inference in a grid is computationally prohibitive due to a large treewidth, various approximate inference methods have been proposed for parameter learning. In this work, we consider two simple approximations to the conditional likelihood (CL): mean field (MF) and loopy belief propagation (LBP). Vishwanathan *et al*. [8] promote use of stochastic gradient methods for efficient CRF learning instead of the more conventional limited memory Broyden-Fletcher-Goldfarb-Shanno (BFGS) method. Of the several methods they compared, they observe that Stochastic Meta-Descent (SMD) was the most effective due to is adaptive annealing schedule. Hence, we use SMD as the optimizer. These parameter learning, inference, and optimizer algorithms were implemented using Kevin Murphy's 2D-CRF toolbox publicly available at [11].

3 Approach

We work with two different image classification tasks. The first task is a two-class problem, where the image regions have to be classified as natural or manmade structures. The images were taken from the Corel database. The second task is a multiclass problem where the image regions are to be classified into sky, grass, cow, building and tree. The images were taken from the Microsoft Research at Cambridge (MSRC) database [13], which also provides hand segmented ground truth. Regions of the image that could not segmented accurately were labeled as void. For both classification tasks, the images were divided into non-overlapping blocks, and each of these blocks formed a site (node) in the graph, for which a classification label was to be inferred.

The task of classifying the image regions into natural and manmade structures using the Corel database has been presented by Kumar *et al*. in [3, 6] where they establish the superiority of CRFs over MRFs for this task. For the first half of the project time-line, our goal was to reproduce these results. For each image site, a 5 dimensional single-site feature vector and a 14 dimensional multi-scale feature vector $\boldsymbol{f}_i(\boldsymbol{x})$ is computed using gradient based features as described in [12], which incorporates the data interaction from neighboring sites. The code for this feature extraction was obtained from [11]

Our main focus for the rest of the project was on the classification task of the MSRC database classification task. We focussed on three main tasks: Feature extraction (extracting features at each pixel in the image), feature aggregation (combining features within a block to produce once feature vector that represents the entire block), and combining binary classifiers via weighted majority voting to achieve multiclass classification. Our approach to each of these is described below.

3.1 Feature extraction

A standard approach in object recognition for deformable objects is to treat the problem as that of texture classification [1, 2]. The characteristics of texture are extracted by filtering the image with several different filters. We design a filter bank similar to that used by Winn *et al.* [1] as shown in Figure 1. It includes a Gaussian filter (smoothed version of the averaging filter), Laplacian of Gaussian filter (LoG, smoothed version of the second derivative filter), and the derivative of Gaussian filters (smoothed versions of the first derivative filter) - all at several different scales (σ). The RGB image is transformed to the CIE L,a,b colorspace in a similar fashion as [1]. We convolve these filters with the image, and the response of a pixel to each of these filters is concatenated to form a feature vector corresponding to that pixel. For most filters, only the L-channel of the image was utilized, however for the gaussian filter all three channels were utilized, similar to [1].

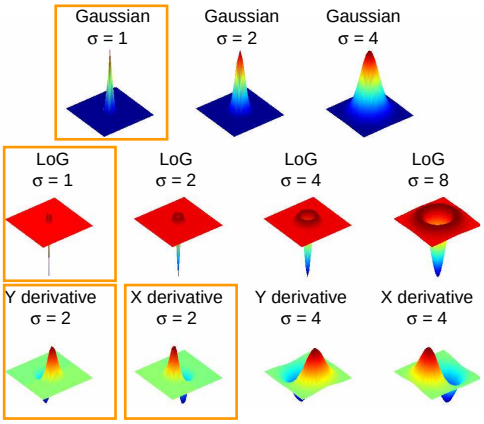


Figure 1: Filter bank used for feature extraction.

Motivated by Kumar *et al.* [6], we extract single-scale features that characterize the pixel locally, as well as multi-scale features that characterize the pixel at several scales of it's neighborhood. The filters highlighted in Figure 1 were used for extracting the 6 dimensional single-scale feature, while all the filters displayed are used to extract a 17 dimmensional multi-scale feature for each pixel. A comparison of the classification accuracy by using single-scale features as opposed to multi-scale features is provided in Section 4.

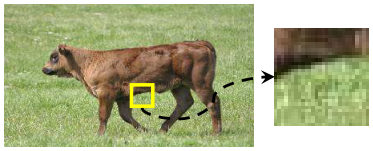
3.2 Feature aggregation

As stated earlier, we divide the image into non-overlapping blocks, and each of these blocks (and not each pixel) is a site to be classified. Hence, having extracted a feature vector at each pixel, the goal is to combine the feature vectors from pixels that share a common block in an appropriate way to represent the entire block.

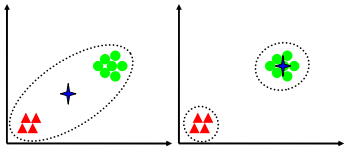
One approach was to assume that the pixels belonging to a block come from the same region, and hence the distribution of the features within a block is unimodal. In this case, the average response within a block is used as the representative response.

However, consider a block as shown in Figure 2 (a). This block contains pixels from grass as well as cow. Modeling this block as a unimodal distribution of features would be inappropriate, and the average response of the block, as illustrated in Figure 2 (b, left), would be meaningless. This is resolved by applying K-means clustering to each block. The appropriate value of K, the number of clusters, is picked by maximizing the minimum

description length (MDL) criterion. Having picked the optimum value of K, the most dominant (largest) cluster was used as the representative cluster, and the average response of this cluster was used as the representative feature vector for the block, as illustrated in Figure 2 (b,right).



(a) A patch covering two textured regions



(b) Multimodal modeling of patch for feature aggregation

Figure 2: Feature aggregation

Again, a comparison between the classification performance of this method and the naive unimodal modeling of each block is provided in Section 4.

3.3 Multiclass classification

Kevin Murphy's tool box [11] only supports a binary classification problem. In order to apply it to a C class problem, we train C binary classifiers that identify the corresponding class as positive, and the rest as negative. For a given test instance, the outputs of these binary classifiers are combined using weighted majority voting. Two different strategies were employed to assign weights to the classifiers. One was the strength of the classifiers, as quantified by it's performance on the training data, and the other was the confidence of the classifier as quantified by the belief of that classifier for that instance (node). In both cases, if more than one classifier assigns a positive label to an instance, the label corresponding to the classifier with the highest weight is assigned to that label. In both cases, if none of the classifiers assign a positive label to the instance, we assign it to a *reject* class. This is particularly relevant when we experiment with different number of classes. It should be noted that this is not equivalent to the classification system rejecting an instance and not making a decision for that instance, which does not play a role in the classification accuracy. A comparison of the classification performance obtained using the strength of the classifier as weights as opposed to using the confidence of the classifier during weighted majority voting is provided in Section 4.

All possible combinations of the above design choices were experimented with.

4 Experimental results

As stated earlier, we applied CRFs for image region classification to two databases.

The Corel database contained 108 images for training and 129 images for testing. The same split as that used by Vishwanathan *et al.* was used so that the results obtained can be compared. The images were 256×384 pixels and were divided into non-overlapping blocks of 16×16 pixels. An example of the results obtained on these images is shown in Figure 3. The regions classified as natural are labeled in black, while the regions classified as manmade structures are marked in white. The test error, as measured by the proportion of misclassifications was found to be 0.12 using MF for inference. The results obtained are similar to those reported by Kumar *et al*. [6] and Vishwanathan *et al*. [8].

For the MSRC database, 53 images were used for training and 49 images were used for testing. This split was made randomly, maintaining consistent distribution of classes in

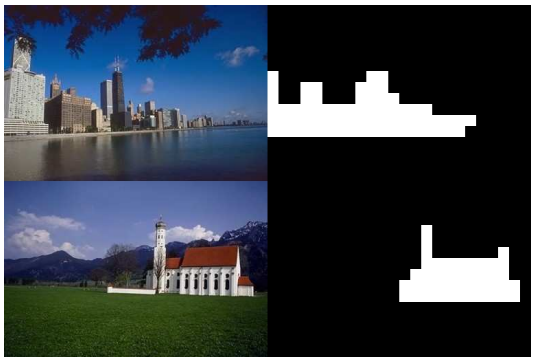


Figure 3: Example results on Corel images

both sets. The images were 312×208 pixels and were broken down into blocks of 13×13 pixels. Different number of classes were considered, where the number of classes was increased by adding classes in the descending order of their performance on the training set. As stated earlier, for each of these, all possible combinations for the different design choices were experimented with. These were repeated for five different splits of training and testing data sets.

The first analysis from the above experiments was to analyze the effect of each of our design choices on the classification accuracy. For instance, the performance using single-scale features was compared to that using multi-scale features, averaged across all possible choices for the other factors such as the feature aggregation techniques, weights used for weighted majority voting in combining classifiers, the inference engine, and five random splits of the data into training and testing. The results obtained are summarized in Figure 4. The average performance and the 95% confidence intervals are shown. The results reported are considering only two classes, however similar trends were observed with more classes.

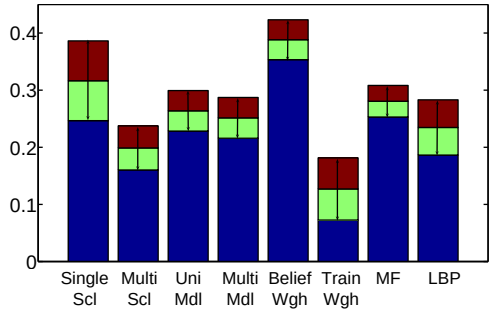


Figure 4: Effect of different design choices on test classification error

It can be seen that multi-scale feature extraction perform better than single-scale with statistical significance. This indicates that incorporating information from neighboring sites while making decision about a particular site enhances the classification accuracy. Modeling each block with a multimodal distribution during feature aggregations does not enhance the performance. This may be due to the fact that the blocks are fairly small, and hence most blocks have homogeneous texture. Using the strength of the classifiers as weights in majority voting performs significantly better than using the confidence of the classifiers. This is because, especially with LBP, the actual values of the beliefs are over-confident and

hence not reliable. For inference, LBP has a better average accuracy then MF, however it has a high variance as compared to MF, and hence the difference was not found to be statistically significant. Based on the above analysis, the following design choices were made: multi-scale features, modelling the blocks to be unimodal distribution of the features (for computational simplicity), using the training performance as weights to combine classifiers, and LBP for parameter learning and inference.

Having made these choices, we observe the effect of added classes on the classification accuracy. The graph of the classification error vs. number of classes considered is shown in Figure 5. It can be seen that the classification error increases quite drastically as the number of classes increase. Apart from the added complexity due to more classes, the order in which the classes are added incrementally - in decreasing order of their classification accuracy on the training data, also adds to the degradation in performance. However, since the training performance is used as weights during weighted majority voting for combining the classifiers, this order is a natural choice. Figure 5 shows a comparison between the performance of CRFs and two naive baselines where all regions were given the same label (1) assuming uniform prior among classes and (2) learning the prior from the training data. This shows that CRFs are providing meaningful labels.

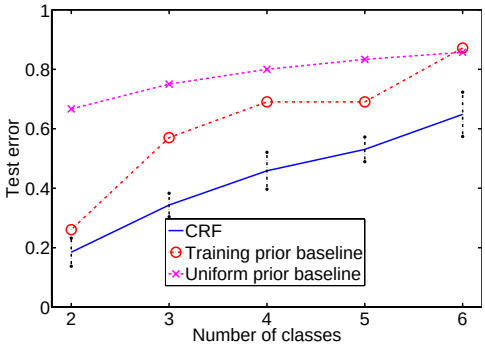


Figure 5: Effect of number of classes on test classification error

Classification results on sample images are shown in Figure 6. Three classes were considered - grass, cow and sky. The green labels correspond to grass, the red regions to cow, the blue regions to sky and the black regions to the *reject* class. It can be seen that the regions are classified fairly accurately. The region corresponding to the building has indeed been rejected by the grass, cow and sky classifiers. It can be seen that the segmentation obtained is rather coarse, not only due to the coarseness introduced by the blocking of images, but also due to the smoothness constraints in CRFs.

5 Conclusion

We used a discriminative framework, CRFs, that allow for incorporation of spatial dependencies in images to classify image regions based on their texture. We used mean field and loopy belief propagation for parameter learning and approximate inference. We experiment with two image classification tasks, a binary classification problem to distinguish between natural and manmade structures, and a multiclass problem involving trees, grass, cow, building and sky. We mainly focused on feature extraction, feature aggregation and techniques to use binary classifiers for the multiclass classification problem. The effects of the different design choices as well as increasing number of classes on the classification performance was analyzed. While using CRFs provides meaningful results (as compared to

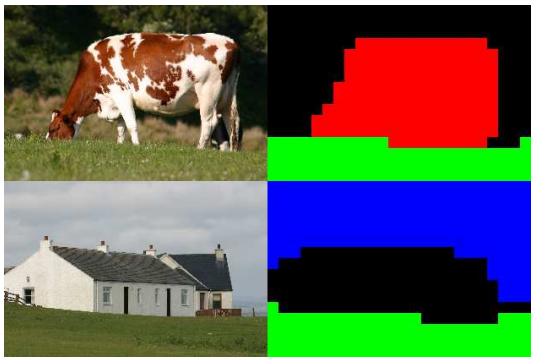


Figure 6: Example results on MSRC images

extremely naive approaches), the classification errors were high, especially as more classes are added to the problem. Potential scopes of improvement include using a validation set of images instead of training images to compute the weights of the classifiers (if more data is available), experimenting with different orders of classes being added, performing multi-factor statistical analysis to determine the optimum selection of design choices as opposed to the single-factor analysis, learning the mapping of the beliefs of classifiers to a more reliable estimate of the classifier's confidence that also incorporates the strength of the classifiers, monitoring the time taken for learning and inference as the dimensionality of the features increases, extending the CRF toolbox to truly handle multiclass problems, and comparing the performance of the CRFs with other baselines such as logistic regression.

References

[1] J. Winn, A. Criminisi, and T. Minka. *Object Categorization by Learned Universal Visual Dictionary*. ICCV 2005

[2] S. Yu, and J. Shi. *Object-Specific Figure-Ground Segregation.* CVPR 2003

[3] S. Kumar, and M. Hebert. *Discriminative Random Fields: A Discriminative Framework for Contextual Interaction in Classification*. ICCV 2003

[4] H. Wallach. *Conditional Random Fields: An Introduction*. Tech Report, University of Pennsylvania, 2004

[5] J. Lafferty, A. McCallum, and F. Pereira. *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*. ICML 2001

[6] S. Kumar, and M. Hebert. *Discriminative Fields for Modeling Spatial Dependencies in Natural Images*. NIPS 2004

[7] S. Li. *Markov Random Field Modeling in Image Analysis*. Springer-Verlag, 2001

[8] S. Vishwanathan, N. Schraudolph, M. Schmidt, and K. Murphy. *Accelerated Training of Conditional Random Fields with Stochastic Gradient Methods*. ICML 2006

[9] Y. Weiss. *Comparing the Mean Field Method and Belief Proagation for Approximate Inference in MRFs*. Advanced Mean Field Methods, MIT Press.

[10] V. Kolmogorov. *Convergent Tree-reweighted Message Passing for Energy Minimization*. Tech Report, Microsoft Research, 2004

[11] http://www.cs.ubc.ca/ murphyk/Software/CRF/crf.html

[12] S. Kumar, and M. Hebert. *Man-Made Structure Detection in Natural Images using a Causal Multiscale Random Field*. CVPR 2003

[13] Microsoft Research, Cambridge. http://research.microsoft.com/vision/cambridge/recognition/