AN ABSTRACT OF THE THESIS OF

Travis Moore for the degree of Master of Science in Computer Science presented on November 21, 2013.

Title: Incorporating Labeled Features into Image Classification using Generalized Expectation

Abstract approved: _____

Weng-Keen Wong

Image classification is a difficult problem, often requiring large training sets to get satisfactory results. However this is a task that humans perform very well, and incorporating user feedback into these learning algorithms could help reduce the dependency on large amounts of labeled training data. This process has already been leveraged in text classification, through the incorporation of labeled features. Labeling features provides a much more informative form of feedback than existing vision feedback systems like active learning and relevance feedback. In this paper, I adapt the Generalized Expectation Criteria to incorporate labeled features into the more complex CRF model used for images. Experiments are performed using oracle selected features as a first step towards showing the potential benefits of this kind of user feedback for image classification.

Incorporating Labeled Features into Image Classification using Generalized Expectation

by

Travis Moore

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented November 21, 2013
Commencement June 2014

Master of Science thesis of Travis Moore presented on November 21, 2013.

APPROVED:

Major Professor, representing Computer Science

Director of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Travis Moore, Author

ACKNOWLEDGEMENTS

I would like to acknowledge Mohamed Amer who provided implementations of the various feature extractors used in these experiments. I would also like to thank Dr. Wong and Dr. Todorovic for their help in completing this project, and of course my parents for their continued support of my academic career.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1: Introduction

Image classification is the task of assigning a classification label (or set of labels) to an image based on the objects and scene it depicts. This can take many different forms, depending on the application. Algorithms can detect the presence or absence of objects in an image, denote the specific location of an object, retrieve other images that contain the same or similar objects, or even a combination of these. Whatever the application, image classifcation has proven to be a difficult problem, often requiring a large amount of training data to achieve satisfactory results. While it is relatively easy to obtain a large database of images, labeling these images is still a time consuming process. With the growing uses of computer vision in surveillance and security footage, facial recognition, video analysis, and object recognition, it would be advantageous to find a way around the need for large amounts of labeled data.

Despite the difficulties of doing image classification for machines, humans are very good at it. Humans perform this task so well, that in [24] they use human stand-ins for various components of their algorithm to identify how much it can be improved in future work. Identifying the objects that we see is something we do constantly every day. We are the experts in this domain, and any knowledge that we could pass on to the computers that perform this task could improve accuracy without the need for more and more labeled data.

User-in-the-loop systems like active learning and relevance feeback already exist for image classifications [15, 34, 33, 19, 17, 20, 36, 1], but the user feedback obtained from these systems is rather shallow. With these approaches users can tell the computer *what* an object is or is not, but not *why*. For example, users can tell the system whether or not an object is a car, but not the qualities that make up a car (such as having four wheels, a windshield, metallic surface, etc). Telling the computer why an object is what it is has been partially addressed in [3, 26] through the use of relative attributes. Here the user feedback relates two classes through an attribute, such as saying that a meadow should be more "open" then a forest. This kind of relative information has been proven to be beneficial for image classification, and it stands to reason that a more direct input,

such as identifying features that objects possess, could improve accuracy as well.

Providing user feedback on features is something that has already been shown to be beneficial in text classification [9], another domain where humans have a wealth of experience to impart upon the system. Yet, this approach remains under utilized for the task of image classification. In this paper, I take the first steps in showing that labeled features can improve image classification algorithms, specifically in the case when little training data is available.

To incorporate this feature feeback into parameter learning, I adapt a technique from text classification known as generalized expectation [23, 10]. The generalized expection criteria (or GE) is a regularization term that penalizes the model for deviating from a target distribution. For the incorporation of labeled features, the target distribution is defined as a skewed distribution that highly favors the provided feature labeling. GE was chosen for the general nature of its formulation. While in this paper I work with feedback at the feature level, the GE term could easily be extended in future work to use more abstract feedback, such as high level attributes or user-defined feature combinations previously unknown to the algorithm. This is because GE works on the conditional distribution where a given feature is present. So as long as we can design a detector for a feaure or attribute, we can use it in GE.

Due to the greater complexity present in image data, the original GE formulation must be extended to work with CRFs with higher connectivity. The derivation of this is shown in section 3.3, and is done in a way so as to be easily extendable to graphical models with relatively large connectivity.

The contributions of this paper are two fold: 1. This paper demonstrates a proof of concept implementation for incorporating labeled features into image classification algorithms. 2. This paper expands the original formulation of generalized expectation criteria to be applicable to more complex models. The hope is that with the ground work laid by this paper, we can start to take steps towards incorporating labeled features and richer forms of feedback into image classification algorithms, and making a system that is as accurate as we are without being dependent on huge amounts of labeled data.

Chapter 2: Related Work

With the challenges presented in image classification, there have been many different methods proposed to reduce the number of training examples needed, often by including some sort of external knowledge. Semi-supervised learning, like the method presented in [5] is one such approach. These methods seek to leverage the information available in unlabeled images, or in the case of [28], unlabeled videos. Other methods include adding informative priors [30] and domain knowledge [25]. These are both one-shot approaches, where the external information is collected before training. Under these systems, there is no way to tailor results to specific users, or incorporate additional information *post-deployment*.

Relevance feedback is a method that works directly with users as part of a feedback loop for the task of image retrieval [33][19]. Here the user gives feedback on query responses in an attempt to leverage additional information outside of the training data. This kind of feedback is tuned to the user and collected after the algorithm is deployed. Recent work [17] prompts users for relative attributes, where the machine asks if the intended query should have more or less of a specific quality than an example image. These are abstract qualities, such as a scene being open or crowded, that are human understandable and require specific machine detectors. In contrast, the feedback used in this paper is concrete features labels, where objects are specified as having certain features. For example, with relative attributes we could specify that cars are shinier than cows, where as with feature labels we can say that cars *are* metalic. Incorporating these feature labels for abstract qualities is a subject for future work.

Active learning is another user feedback approach that seeks to get the most out of labeled data by working directly with the user. This approach was explored in relation to image classification in [15] and [35], as well as other approaches that work at the pixel level [34] or region level [31] of images. Other varients include adaptive active learning [20], crowd sourced active learning [36], and learning action detectors [1]. Most closely related to feature labeling is the work in [26] and [3]. In these papers, when the user corrects the machine's labeling, they also specify why the label should be different, using

the language of relative attributes. For example, if the machine presents an image of a meadow and suggests that it is a forest, the user can correct it by saying that the image is "too open" to be a forest. Like the relevance feedback approach in [17], the attributes are used in a relative way, and the computer can conclude that other images that are more or equally "open" as the presented image are also not forests. Again, the main difference with the work presented here is that the labeling is done in a relational way. Rather that the user specifying the attributes of a forest, they identify how it is different from a meadow.

Outside of computer vision, there have been other attempts to incorporate external knowledge into structured learning problems. In [4] the external knowledge is in the form of a taxonomy. In a more general approach, [6] and [7] use knowledge for structured learning in the form of contraints. These constraints guide the learning process, often in a semi-supervised setting. This is similar to the use of Generalized Expectation Criteria.

The Generalized Expectation Criteria was introduced by Druck, Mann, and McCallum in [23] as a general method for incorporating preferences into learning model parameters. Although GE is capable of influencing models to include constraints of virtually any nature, the majority of this work revolved around using GE as a way to incorporate labeled features into an algorithm. This proceedure forms the baseline of the approach taken in this paper. The authors applied this regularizer to logistic regression [10] and conditional random fields [22] as a way of improving classification accuracy when few training instances were available. An active learning approach was presented as well in [11], again using conditional random fields. The formulation done for conditional random fields was done for linear chain CRFs, not general ones like in this paper, and only used labeled features for single nodes, rather than pairs of nodes as well.

Using GE to incorporate labeled features lends itself naturally to text based data and natural language processing, which is where the majority of GE approaches have been done. This includes document classification and annotation in the 2008 papers, as well as more recent approaches in cross-language named entity recognition [38] and language labeling in mixed-language documents [16]. This paper represents the first use of generalized expectation for the use of image classification.

There are of course other methods for labeling features besides GE. Raghavan and Allan proposed a method for using labeled features in support vector machines [27], and [9] incorporates labeled features into a logistic regression model. The latter method can

be considered a special case of GE, as can other methods that use maximum entropy or transfer learning [23]. The end goal for a feature labeling algorithms is to solicit the labels from users. With the domain of image classification to consider, these labels could be on features, learned attributes, or even user-defined attributes. The formulation of GE is general enough to handle all these cases, thus allowing the algorithm to easily scale with future work. This paper represents a first step in this process for the vision domain, using feature labels obtained from an oracle as a proof of concept.

Chapter 3: Methodology

In this section, I will go over the formulations for the algorithms presented in this thesis, along with their specific implementations. I will start with an overview of Conditional Random Fields, along with notes on the vision specific CRF used in experiments. Then I will discuss the generalized expectation regularizer, both in terms of a general formulation and an approximate version for use with more complex data. I will conclude with some notes about the data sets and experimental methodology.

3.1 Conditional Random Fields

A Conditional Random Field (or CRF) is a discriminative classification algorithm. It is a more general application of logistic regression, where we allow multiple class labels for a single data instance. In general, the CRF model consists of a set $\vec{X}$ of observed variables, which we can derive features from, a set $\vec{Y}$ of output variables which we try to predict, and an edge set $\vec{E}$ denoting the connections between variables.

A CRF models the conditional distribution $P(\vec{Y}|\vec{X})$ as a product of factors, where each factor is the exponent of a weighted sum of features over a subset of $\vec{X}$ and $\vec{Y}$.

$$P(\vec{Y}|\vec{X}) = \frac{1}{Z(\vec{X})} \prod_{A \in \Psi} exp(\sum_{k=1}^{K(A)} \lambda_{Ak} f_{Ak}(\vec{y}_A, \vec{x}_A)) \tag{3.1}$$

Here Ψ is the set of all factors, f is a feature function for a given factor A and feature index k, $K(A)$ is the number of features for factor A, and λ is a feature weight. Each factor A represents a subset of related variables from $\vec{X}$ and $\vec{Y}$, denoted as $\vec{y_A}, \vec{x_A}$. $Z(\vec{X})$ is a normalization term. The goal of training a CRF is to learn the parameter values λ.

In practice, it is often desirable to have several factors share the same parameter values, either to enforce similar distributions or to simplify the model by reducing the number of parameters needed. When this is the case we use clique templates, sets of factors that have the same values of λ, in the model formulation.

$$P(\vec{Y}|\vec{X}) = \frac{1}{Z(\vec{X})} \prod_{c \in \mathbf{C}} \prod_{\Psi_c \in c} exp(\sum_{k=1}^{K(c)} \lambda_{ck} f_{ck}(\vec{y}_{\Psi_c}, \vec{x}_{\Psi_c})) \tag{3.2}$$

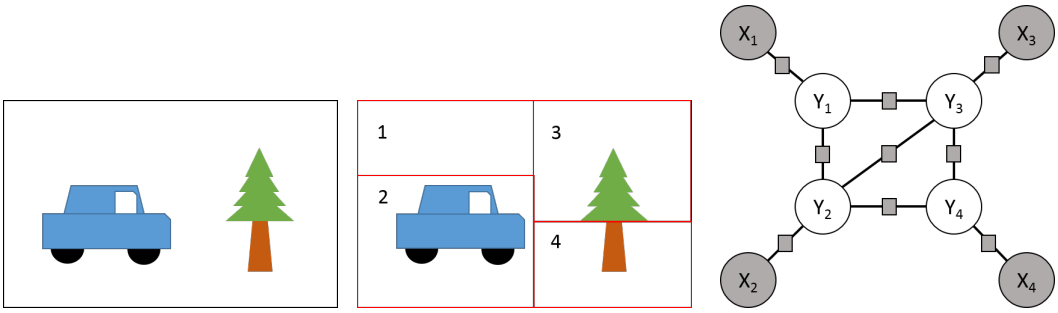
Here $\mathbf{C}$ is the set of all clique templates, c a specific clique template, and Ψ_c is a factor (or clique) in c. This is different from the previous formulation in that the λ and f terms are indexed by their clique template, rather than factor. Cliques are often defined as fully connected subgraphs, with a size equal to the number of nodes in the subgraph. For example, a clique of size one is a single node, a clique of size two is two nodes connected together, and a clique of size three is a cycle of three nodes. If we define our clique templates to be over clique sizes of one and two, then we will have a factor for every individual node, and a factor for every pair of connected nodes.

3.2 CRF for Image Classification

For the purpose of image classification, we define a specific type of CRF based on the segmentation graph (see Figure 3.1). Each segment of an image has several descriptors extracted from it, namely SIFT[21], HOG[8], color histogram[32], and shape context[2]. These vector descriptors are clustered using k means (k set at 200), with each cluster index representing a binary feature. For example, if a segment has a SIFT vector s, and it is clustered into group j, then we say that the feature $SIFT_j$ is present in the segment. Each segment has only one SIFT, HOG, color, and shape feature present, making for a very sparse feature vector representation. These features make up the set $\vec{X}$ in the model, and the object labels of each segment make up the set $\vec{Y}$. For this paper, we assume that each segment or region has only one label.

In the graphical model, each set of segment features is connected to its segment label. We also define two segment labels to be connected if the segments are adjacent to each other. This kind of connection in the graph allows adjacent segments to influence each other during inference, which captures the important spatial relationships and information found in image data (for example, a cow would be more likely to be adjacent to grass than to be adjacent to an airplane). This also makes the image data much more complex, as we have cycles in the graph (making exact inference techniques impractical) and a potential for cliques of many different sizes.

For our purposes, we define our clique templates to be over clique sizes of one and



(a) Image (b) Image Segmentation (c) CRF Graph

Figure 3.1: An example image (a), its segmentation (b), and the corresponding CRF in graphical notation (c). The Y nodes correspond to segment labels, and the X nodes represent observed features extracted from segments. The lines and boxes represent factors in the CRF, which occur between label and feature nodes (unary factors) and pairs of label nodes (binary factors). Note that there is no connection between Y_1 and Y_4, as segments 1 and 4 do not share any edges. Here the ground truth labeling of $\{Y_1, Y_2, Y_3, Y_4\}$ is $\{background, car, tree, tree\}$.

two. This means that the CRF will have a factor for every segment, and a factor for every pair of adjacent segments. Using these two clique templates allows our CRF to still capture some spatial information while remaining tractable to learn. The features for factors over single nodes will be the ones extracted from the corresponding segments (called unary features), and the features for factors over pairs of nodes will be the union of their unary features.

Knowledge is incorporated into the CRF using the framework of generalized expectation, or GE. GE is a regularizer based on labeled features. That is, we provide a list of features and the classes that they should belong to, and the GE regularizer pushes the CRF parameters to match this input. Training the CRF is done using gradient ascent, which involves calculating the gradient. Calculating the gradient of the likelihood term is a solved problem, however finding the gradient of the GE regularizer is trickier, and is explained further in the next section.

3.3 GE Regularizer

The generalized expectation regularization term introduced by Druck et al [10], is of the form

$$-GE(P^T, \hat{P}_\theta) = -\sum_{f \in F} KL(P^T(\vec{y}|x_f > 0)||\hat{P}_\theta(\vec{y}|x_f > 0)) \tag{3.3}$$

where P^T is a target distribution that the algorithm is trying to match, $\hat{P}_\theta$ is an empirical distribution based on available data instances, F is the set of labeled features, and KL stands for Kullback Leibler divergence. This regularizer is simply the distance between our target distribution, which is user defined, and the empirical distribution, which is estimated using current model parameters. The distance is conditioned on the presence of a set of label features which have been designated as highly informative. We follow a similar formulation in this paper, but with considerations taken to apply to a general CRF graph.

For our formulation, labeled features consist of the set $[x_f, y_f, C_f]$, where x_f is the feature index, y_f is the label, and C_f is the clique template of the feature. The dimensions of y_f will depend on the size of the clique template. If the feature belongs to cliques of size one, then y_f is a single label. If it belongs to cliques of size two, y_f is a pair of labels. In line with this, we define our target and empirical distributions based on the average distribution of all the factors in C_f.

$$\hat{P}_\theta(\vec{y}_{C_f} = \vec{y}_f|x_f > 0) = \frac{1}{Z_f} \sum_{\vec{x} \in U} \sum_{\psi \in \mathbf{C}_f} I(x_f \in \vec{x}_\psi) P_\theta(\vec{y}_\psi = \vec{y}_f|\vec{x}) \tag{3.4}$$

$$Z_f = \sum_{\vec{x} \in U} \sum_{\psi \in \mathbf{C}_f} I(x_f > 0) \tag{3.5}$$

Here Z_f is a normalization term, U a pool of unlabeled data used to estimate the model distribution, and $P_\theta(\vec{y}_{\psi_f} = \vec{y}_f|\vec{x})$ is found through marginalization of the CRF using its current parameter values. Note that in this distribution we are looking at a subset of the labels, denoted as $\vec{y}_{C_f}$. Since each clique template is a set of factors with the same parameter values, we calculate one empirical distribution for each clique template, averaged over all these factors. This is to stay consistent with the clique template representation, where factors in a template are regarded as different draws from the same underlying distribution. The marginalization term, $P_\theta(\vec{y}_\psi = \vec{y}_f|\vec{x})$, can

be further expanded using the CRF formulation.

$$P_\theta(\vec{y}_\psi = \vec{y}_f|\vec{x}) = \sum_{\vec{y}:\vec{y}_\psi=\vec{y}_c} P_\theta(\vec{y}_\psi = \vec{y}_f, \vec{y}_{-\psi} = \vec{y}_-|\vec{x})$$

$$= \sum_{\vec{y}:\vec{y}_\psi=\vec{y}_c} \frac{1}{Z(\vec{x})} \prod_{c\in\mathbf{C}} \prod_{\psi_c\in c} \exp(\sum_{k=1}^{K} \lambda_{(c,k,\vec{y}_t)} I(x_k \in \vec{x}_{\psi_c}) I(\vec{y}_{\psi_c} = \vec{y}_t)) \tag{3.6}$$

$$Z(\vec{x}) = \sum_{\vec{y}} \prod_{c\in\mathbf{C}} \prod_{\psi_c\in c} \exp(\sum_{k=1}^{K} \lambda_{(c,k,\vec{y}_t)} I(x_k \in \vec{x}_{\psi_c}) I(\vec{y}_{\psi_c} = \vec{y}_t)) \tag{3.7}$$

Where $\vec{y} : \vec{y}_\psi = \vec{y}_c$ is the vector of all Y's except $\vec{y}_\psi$. Note that we have defined our feature function for the exponent simply as a pair of indicator functions for the presence of x_k in $\vec{x}_{\psi_c}$ and the value of $\vec{y}_{\psi_c}$. The target distribution is created using a Schapire distribution[29] over the possible labels[10] of the assigned clique template, with the majority of the weight being put on the label assigned to the feature, y_f.

$$P^T(\vec{y}_{C_f} = y_f|x_f > 0) = q_{max} \tag{3.8}$$

$$P^T(\vec{y}_{C_f} \neq y_f|x_f > 0) = \frac{1 - q_{max}}{|\vec{y}_{C_f} : \vec{y}_{C_f} \neq y_f|} \tag{3.9}$$

q_{max} is generally set to something high, like 0.9 or 0.8, but is not overly sensitive to the exact value. For the results reported here, I use the value 0.9.

In order to perform gradient ascent on the regularized CRF, we will also need to calculate the partial derivative of our regularizer. The partial derivative is with respect to the parameter values, which are indexed by clique template (C_t), feature index (j), and factor label ($\vec{y}_L$).

$$\frac{\partial}{\partial\lambda_{(C_t,j,\vec{y}_L)}}(-\triangle(P^T, \hat{P}_\theta)) =$$

$$\frac{\partial}{\partial\lambda_{(C_t,j,\vec{y}_L)}}[-\sum_{f\in F}\sum_{\vec{y}_c} P^T(\vec{y}_{C_f} = \vec{y}_c|x_f > 0) \log(\frac{P^T(y_{C_f} \vec{=} \vec{y}_c|x_f > 0)}{\hat{P}_\theta(\vec{y}_{C_f} = \vec{y}_c|x_f > 0)})] \tag{3.10}$$

The KL divergence term is expanded out, with the summation over labels restricted to the clique template for each labeled feature. Remember that in our labeled features,

a y label is assigned along with a clique template. Thus we only worry about the feature being present within the specified cliques.

$$= \frac{\partial}{\partial \lambda_{(C_t,j,\vec{y}_L)}}[-\sum_{f \in F}\sum_{\vec{y}_c} P^T(\vec{y}_{C_f} = \vec{y}_c | x_f > 0)$$

$$[\log(P^T(\vec{y}_{C_f} = \vec{y_c}|x_f > 0)) - \log(\hat{P}_\theta(\vec{y}_{C_f} = \vec{y}_c|x_f > 0))]] \tag{3.11}$$

The terms without $\hat{P}_\theta$ cancel out, since they do not depend on λ, leaving us with

$$\frac{\partial}{\partial \lambda_{(C_t,j,\vec{y}_L)}} \sum_{f \in F}\sum_{\vec{y}_c} P^T(\vec{y}_{C_f} = \vec{y}_c|x_f > 0) \log[\hat{P}_\theta(\vec{y}_{C_f} = \vec{y}_c|x_f > 0)] = \tag{3.12}$$

$$\sum_{f \in F}\sum_{\vec{y}_c} P^T(\vec{y}_{C_f} = \vec{y}_c|x_f > 0) \frac{\partial}{\partial \lambda_{(C_t,j,\vec{y}_L)}} \log[\hat{P}_\theta(\vec{y}_{C_f} = \vec{y}_c|x_f > 0)] = \tag{3.13}$$

$$\sum_{f \in F}\sum_{\vec{y}_c} \frac{P^T(\vec{y}_{C_f} = \vec{y}_c|x_f > 0)}{\hat{P}_\theta(\vec{y}_{C_f} = \vec{y}_c|x_f > 0)} \frac{\partial}{\partial \lambda_{(C_t,j,\vec{y}_L)}} \hat{P}_\theta(\vec{y}_{C_f} = \vec{y}_c|x_f > 0) = \tag{3.14}$$

$$\sum_{f \in F}\sum_{\vec{y}_c} \frac{P^T(\vec{y}_c|x_f > 0)}{\hat{P}_\theta(\vec{y}_c|x_f > 0)} \frac{\partial}{\partial \lambda_{(C_t,j,\vec{y}_L)}} [\frac{1}{Z_f} \sum_{\vec{x} \in U}\sum_{\psi \in C_f} I(x_f \in \vec{x}_\psi) P_\theta(\vec{y}_\psi = \vec{y}_c|x_f > 0)] = \tag{3.15}$$

$$\sum_{f \in F}\sum_{\vec{y}_c} \frac{P^T(\vec{y}_c|x_f > 0)}{\hat{P}_\theta(\vec{y}_c|x_f > 0)} \frac{1}{Z_f} \sum_{\vec{x} \in U}\sum_{\psi \in C_f} I(x_f \in \vec{x}_\psi) \frac{\partial}{\partial \lambda_{(C_t,j,\vec{y}_L)}} P_\theta(\vec{y}_\psi = \vec{y}_c|x_f > 0) = \tag{3.16}$$

$$\sum_{f \in F}\sum_{\vec{y}_c} \frac{P^T(\vec{y}_c|x_f > 0)}{\hat{P}_\theta(\vec{y}_c|x_f > 0)} \frac{1}{Z_f} \sum_{\vec{x} \in U}\sum_{\psi \in C_f} I(x_f \in \vec{x}_\psi) \frac{\partial}{\partial \lambda_{(C_t,j,\vec{y}_L)}}$$

$$[\sum_{\vec{y}:\vec{y}_\psi = \vec{y}_c} \frac{1}{Z(\vec{x})} \prod_{c \in \mathbf{C}} \prod_{\psi_c \in c} \exp(\sum_{k=1}^{K} \lambda_{(c,k,\vec{y}_t)} I(x_k \in \vec{x}_{\psi_c}) I(\vec{y}_{\psi_c} = \vec{y}_t))] \tag{3.17}$$

The term $I(\vec{y}_{\psi_c} = \vec{y}_t)$ in the above equation is to specify the value of $\vec{y}_{\psi_c}$, since the parameters are indexed by y values. This will be important to keep track of later on.

The partial derivative now needs to be applied to $\frac{1}{Z(\vec{x})}$ and the exponential term. Using the product rule, we can look at these two calculations separately.

$$\frac{\partial}{\partial\lambda_{(C_t,j,\vec{y}_L)}}[\sum_{\vec{y}:\vec{y}_\psi=\vec{y}_c}\frac{1}{Z(\vec{x})}\prod_{c\in\mathbf{C}}\prod_{\psi_c\in c}\exp(\sum_{k=1}^{K}\lambda_{(c,k,\vec{y}_t)}I(x_k\in\vec{x}_{\psi_c})I(\vec{y}_{\psi_c}=\vec{y}_t))] = \tag{3.18}$$

$$\sum_{\vec{y}:\vec{y}_\psi=\vec{y}_c}\frac{1}{Z(\vec{x})}\frac{\partial}{\partial\lambda_{(C_t,j,\vec{y}_L)}}[\prod_{c\in\mathbf{C}}\prod_{\psi_c\in c}\exp(\sum_{k=1}^{K}\lambda_{(c,k,\vec{y}_t)}I(x_k\in\vec{x}_{\psi_c})I(\vec{y}_{\psi_c}=\vec{y}_t))]+$$

$$\sum_{\vec{y}:\vec{y}_\psi=\vec{y}_c}\frac{\partial}{\partial\lambda_{(C_t,j,\vec{y}_L)}}[\frac{1}{Z(\vec{x})}]\prod_{c\in\mathbf{C}}\prod_{\psi_c\in c}\exp(\sum_{k=1}^{K}\lambda_{(c,k,\vec{y}_t)}I(x_k\in\vec{x}_{\psi_c})I(\vec{y}_{\psi_c}=\vec{y}_t)) = \tag{3.19}$$

$$\sum_{\vec{y}:\vec{y}_\psi=\vec{y}_c}\sum_{\psi_a\in C_t}[I(x_j\in\vec{x}_{\psi_a})I(\vec{y}_{\psi_a}=\vec{y}_L)]\frac{1}{Z(\vec{x})}\prod_{c\in\mathbf{C}}\prod_{\psi_c\in c}\exp(\sum_{k=1}^{K}\lambda_{(c,k,\vec{y}_t)}I(x_k\in\vec{x}_{\psi_c})I(\vec{y}_{\psi_c}=\vec{y}_t))+$$

$$\sum_{\vec{y}:\vec{y}_\psi=\vec{y}_c}\frac{\partial}{\partial\lambda_{(C_t,j,\vec{y}_L)}}[\frac{1}{Z(\vec{x})}]\Psi \tag{3.20}$$

Here Ψ is used to represent the un-normalized CRF equation. Some explanation on the previous step: in calculating the partial of our product of exponentials, only some of the terms will contain the parameter of interest. Recall that λ is indexed by C_t, which specifies either unary or binary clique templates, the feature index x_j, and the factor label $\vec{y}_L$. Only the terms in the product with matching λs will count, while the rest will drop off. The derivative of each term in the remaining product will just be the indicator functions in the exponential. Using the product rule, we would end up with a summation of these indicator functions for as many factors as are in C_t, indicated as the sum of $\psi_a \in C_t$.

$$\sum_{\vec{y}:\vec{y}_\psi=\vec{y}_c}\sum_{\psi_a\in\mathbf{C}_t}[I(x_j\in\vec{x}_{\psi_a})I(\vec{y}_{\psi_a}=\vec{y}_L)]\frac{1}{Z(\vec{x})}\prod_{c\in\mathbf{C}}\prod_{\psi_c\in c}\exp(\sum_{k=1}^{K}\lambda_{(c,k,\vec{y}_t)}I(x_k\in\vec{x}_{\psi_c})I(\vec{y}_{\psi_c}=\vec{y}_t))+$$

$$\sum_{\vec{y}:\vec{y}_\psi=\vec{y}_c} \frac{\partial}{\partial\lambda_{(C_t,j,\vec{y}_L)}}[\frac{1}{Z(\vec{x})}]\Psi$$

$$= \sum_{\vec{y}:\vec{y}_\psi=\vec{y}_c} \sum_{\psi_a\in\mathbf{C}_t} [I(x_j\in\vec{x}_{\psi_a})I(\vec{y}_{\psi_a}=\vec{y}_L)]P(\vec{y}|\vec{x})+$$

$$- \sum_{\vec{y}:\vec{y}_\psi=\vec{y}_c} \frac{1}{Z(\vec{x})^2}\frac{\partial}{\partial\lambda_{(C_t,j,\vec{y}_L)}}[Z(\vec{x})]\Psi \tag{3.21}$$

$$= \sum_{\vec{y}:\vec{y}_\psi=\vec{y}_c} \sum_{\psi_a\in\mathbf{C}_t} [I(x_j\in\vec{x}_{\psi_a})I(\vec{y}_{\psi_a}=\vec{y}_L)]P(\vec{y}|\vec{x})+$$

$$- \sum_{\vec{y}:\vec{y}_\psi=\vec{y}_c} \frac{1}{Z(\vec{x})^2}\frac{\partial}{\partial\lambda_{(C_t,j,\vec{y}_L)}}[\sum_{\vec{y}}\prod_{c\in\mathbf{C}}\prod_{\psi_c\in c}\exp(\sum_{k=1}^{K}\lambda_{(c,k,\vec{y}_t)}I(x_k\in\vec{x}_{\psi_c})I(\vec{y}_{\psi_c}=\vec{y}_t))]\Psi \tag{3.22}$$

$$= \sum_{\vec{y}:\vec{y}_\psi=\vec{y}_c} \sum_{\psi_a\in\mathbf{C}_t} [I(x_j\in\vec{x}_{\psi_a})I(\vec{y}_{\psi_a}=\vec{y}_L)]P(\vec{y}|\vec{x})+$$

$$- \sum_{\vec{y}:\vec{y}_\psi=\vec{y}_c} \frac{1}{Z(\vec{x})^2}[\sum_{\vec{y}}\frac{\partial}{\partial\lambda_{(C_t,j,\vec{y}_L)}}\prod_{c\in\mathbf{C}}\prod_{\psi_c\in c}\exp(\sum_{k=1}^{K}\lambda_{(c,k,\vec{y}_t)}I(x_k\in\vec{x}_{\psi_c})I(\vec{y}_{\psi_c}=\vec{y}_t))]\Psi \tag{3.23}$$

$$= \sum_{\vec{y}:\vec{y}_\psi=\vec{y}_c} \sum_{\psi_a\in\mathbf{C}_t} [I(x_j\in\vec{x}_{\psi_a})I(\vec{y}_{\psi_a}=\vec{y}_L)]P(\vec{y}|\vec{x})+$$

$$-\frac{1}{Z(\vec{x})}[\sum_{\vec{y}}\frac{\partial}{\partial\lambda_{(C_t,j,\vec{y}_L)}}\prod_{c\in\mathbf{C}}\prod_{\psi_c\in c}\exp(\sum_{k=1}^{K}\lambda_{(c,k,\vec{y}_t)}I(x_k\in\vec{x}_{\psi_c})I(\vec{y}_{\psi_c}=\vec{y}_t))]\sum_{\vec{y}:\vec{y}_\psi=\vec{y}_c}\frac{\Psi}{Z(\vec{x})} \tag{3.24}$$

$$= \sum_{\vec{y}:\vec{y}_\psi=\vec{y}_c} \sum_{\psi_a\in\mathbf{C}_t} [I(x_j\in\vec{x}_{\psi_a})I(\vec{y}_{\psi_a}=\vec{y}_L)]P(\vec{y}|\vec{x})+$$

$$-\frac{1}{Z(\vec{x})}[\sum_{\vec{y}}\Psi\sum_{\psi_a\in\mathbf{C}_t}[I(x_j\in\vec{x}_{\psi_a})I(\vec{y}_{\psi_a}=\vec{y}_L)]]P(\vec{y}_\psi=\vec{y}_c|\vec{x}) \tag{3.25}$$

$$= \sum_{\vec{y}:\vec{y}_\psi=\vec{y}_c} P(\vec{y}|\vec{x})\sum_{\psi_a\in\mathbf{C}_t} I(x_j\in\vec{x}_{\psi_a})I(\vec{y}_{\psi_a}=\vec{y}_L)-$$

$$P(\vec{y}_\psi=\vec{y}_c|\vec{x})\sum_{\vec{y}}P(\vec{y}|\vec{x})\sum_{\psi_a\in\mathbf{C}_t} I(x_j\in\vec{x}_{\psi_a})I(\vec{y}_{\psi_a}=\vec{y}_L) \tag{3.26}$$

$$= [\sum_{\vec{y}:\vec{y}_\psi=\vec{y}_c} P(\vec{y}|\vec{x}) - P(\vec{y}_\psi = \vec{y}_c|\vec{x}) \sum_{\vec{y}} P(\vec{y}|\vec{x})] \sum_{\psi_a \in \mathbf{C}_t} I(x_j \in \vec{x}_{\psi_a}) I(\vec{y}_{\psi_a} = \vec{y}_L) \quad (3.27)$$

$$= [\sum_{\vec{y}} P(\vec{y}|\vec{x}) I(\vec{y}_\psi = \vec{y}_c) - P(\vec{y}_\psi = \vec{y}_c|\vec{x}) \sum_{\vec{y}} P(\vec{y}|\vec{x})] \sum_{\psi_a \in \mathbf{C}_t} I(x_j \in \vec{x}_{\psi_a}) I(\vec{y}_{\psi_a} = \vec{y}_L) \quad (3.28)$$

$$= \sum_{\vec{y}} P(\vec{y}|\vec{x}) [I(\vec{y}_\psi = \vec{y}_c) - P(\vec{y}_\psi = \vec{y}_c|\vec{x})] \sum_{\psi_a \in \mathbf{C}_t} I(x_j \in \vec{x}_{\psi_a}) I(\vec{y}_{\psi_a} = \vec{y}_L) \quad (3.29)$$

Plugging this into our full gradient equation, we get

$$\frac{\partial}{\partial \lambda_{(C_t,j,\vec{y}_L)}}(-\triangle(P^T, \hat{P}_\theta)) = \sum_{f \in F} \sum_{\vec{y}_c} \frac{P^T(\vec{y}_c|x_f > 0)}{\hat{P}_\theta(\vec{y}_c|x_f > 0)} \frac{1}{Z_f} \sum_{\vec{x} \in U} \sum_{\psi \in C_f} I(x_f \in \vec{x}_\psi)$$

$$\sum_{\vec{y}} P(\vec{y}|\vec{x}) [I(\vec{y}_\psi = \vec{y}_c) - P(\vec{y}_\psi = \vec{y}_c|\vec{x})] \sum_{\psi_a \in \mathbf{C}_t} I(x_j \in \vec{x}_{\psi_a}) I(\vec{y}_{\psi_a} = \vec{y}_L) \quad (3.30)$$

The problem here is that summing over all $\vec{y}$ is intractable to do for any CRF with more than a few nodes. This comes from having to take the partial derivate of the marginal probability.

One possible solution to this is to use a variational approximation for the distribution of $P_\theta(\vec{y}|\vec{x})$. A simple approximation to make is the Mean Field approximation.

$$P_\theta(\vec{y}|\vec{x}) = \prod_{y_i} Q(y_i) \quad (3.31)$$

Here we are approximating the full distribution with a product of independent distributions over single nodes. These Q distributions are functions of the exponential form (much like the orginal factors in the model) which are as close as possible to the approximated distribution. Each Q is locally optimal if it is of the form

$$Q(y_i) = \frac{1}{Z_i} \exp[\sum_{\phi:\vec{y}_i \in \phi} E_{\vec{y}\backslash\vec{y}_i|Q}(\ln \phi)] \tag{3.32}$$

Where the inner term is the sum of the expectations given Q of the log of all the factors (ϕ) that $\vec{y}_i$ appears in, over the remaining set of variables, denoted as $\vec{y}\backslash\vec{y}_i$. That is, for each factor we sum out the other variables using the expectation with respect to Q. Z_i is a local normalizer.

For the CRF we have defined, there are two kinds of terms in the summation.

$$Q^t(y_i) = \frac{1}{Z_i} \exp[\ln \phi(y_i) + \sum_{\phi(y_i,y_j)} \sum_{y_j} Q^{t-1}(y_j) \ln \phi(y_i, y_j)] \tag{3.33}$$

The first is simply the parameters from the factor for this node. There are no other variables to sum out from this factor, so we can take it's values as they are. The second is a summation over the pairs in which y_i appears, summing out the values of the other nodes using the distribution of Q. Once they are calculated, each Q term ends up being just like a single CRF factor, with new values for the parameters λ.

We can now substitute this approximation in for P_θ when doing the gradiant calculation. We pick up the derivation from equation 3.16.

$$\frac{\partial}{\partial \lambda_{(C_t,j,\vec{y}_L)}}(-\triangle(P^T, \hat{P}_\theta)) =$$

$$\sum_{f \in F} \sum_{\vec{y}_c} \frac{P^T(\vec{y}_c|x_f > 0)}{\hat{P}_\theta(\vec{y}_c|x_f > 0)} \frac{1}{Z_f} \sum_{\vec{x} \in U} \sum_{\psi \in C_f} I(x_f \in \vec{x}_\psi) \frac{\partial}{\partial \lambda_{(C_t,j,\vec{y}_L)}} P_\theta(\vec{y}_\psi = \vec{y}_c|x_f > 0) \approx$$

$$\sum_{f \in F} \sum_{\vec{y}_c} \frac{P^T(\vec{y}_c|x_f > 0)}{\hat{P}_\theta(\vec{y}_c|x_f > 0)} \frac{1}{Z_f} \sum_{\vec{x} \in U} \sum_{\psi \in C_f} I(x_f \in \vec{x}_\psi) \frac{\partial}{\partial \lambda_{(C_t,j,\vec{y}_L)}} Q(\vec{y}_\psi = \vec{y}_c) = \tag{3.34}$$

$$\sum_{f \in F} \sum_{\vec{y}_c} \frac{P^T(\vec{y}_c|x_f > 0)}{\hat{P}_\theta(\vec{y}_c|x_f > 0)} \frac{1}{Z_f} \sum_{\vec{x} \in U} \sum_{\psi \in C_f} I(x_f \in \vec{x}_\psi)$$

$$\frac{\partial}{\partial \lambda_{(C_t,j,\vec{y}_L)}} \frac{1}{Z_f} \exp(\sum_{k=1}^{K} \lambda_{(c,k,\vec{y}_c)} I(x_k \in \vec{x}_\psi) I(\vec{y}_\psi = \vec{y}_c)) \tag{3.35}$$

The exponent above is an expansion of the Q term, which we have defined to be in exponential form. It should be noted that the λ parameters in this exponential do NOT correspond to the parameters of the true CRF model. They are instead the parameters of the approximated Q distributions, making them a combination of the true parameters. Continuing with the derivation,

$$\sum_{f \in F} \sum_{\vec{y}_c} \frac{P^T(\vec{y}_c | x_f > 0)}{\hat{P}_\theta(\vec{y}_c | x_f > 0)} \frac{1}{Z_f} \sum_{\vec{x} \in U} \sum_{\psi \in C_f} I(x_f \in \vec{x}_\psi)$$

$$\frac{\partial}{\partial \lambda_{(C_t,j,\vec{y}_L)}} \frac{1}{Z_f} \exp(\sum_{k=1}^{K} \lambda_{(c,k,\vec{y}_c)} I(x_k \in \vec{x}_\psi) I(\vec{y}_\psi = \vec{y}_c)) =$$

$$\sum_{f \in F} \sum_{\vec{y}_c} \frac{P^T(\vec{y}_c | x_f > 0)}{\hat{P}_\theta(\vec{y}_c | x_f > 0)} \frac{1}{Z_f} \sum_{\vec{x} \in U} \sum_{\psi \in C_f} I(x_f \in \vec{x}_\psi)$$

$$\frac{1}{Z_f} \frac{\partial}{\partial \lambda_{(C_t,j,\vec{y}_L)}} [\exp(\sum_{k=1}^{K} \lambda_{(c,k,\vec{y}_c)} I(x_k \in \vec{x}_\psi) I(\vec{y}_\psi = \vec{y}_c))] +$$

$$\frac{\partial}{\partial \lambda_{(C_t,j,\vec{y}_L)}} [\frac{1}{Z_f}] \exp(\sum_{k=1}^{K} \lambda_{(c,k,\vec{y}_c)} I(x_k \in \vec{x}_\psi) I(\vec{y}_\psi = \vec{y}_c)) \tag{3.36}$$

This is similar to the where we were in equation 3.19. Following similar steps, we can arrive at

$$\frac{1}{Z_f} \frac{\partial}{\partial \lambda_{(C_t,j,\vec{y}_L)}} [\exp(\sum_{k=1}^{K} \lambda_{(c,k,\vec{y}_c)} I(x_k \in \vec{x}_\psi) I(\vec{y}_\psi = \vec{y}_c))] =$$

$$\frac{1}{Z_f} \exp(\sum_{k=1}^{K} \lambda_{(c,k,\vec{y}_c)} I(x_k \in \vec{x}_\psi) I(\vec{y}_\psi = \vec{y}_c)) I(x_j \in \vec{x}_\psi) I(\vec{y}_\psi = \vec{y}_L) \tag{3.37}$$

$$= Q(\vec{y}_\psi = \vec{y}_c) I(x_j \in \vec{x}_\psi) I(\vec{y}_\psi = \vec{y}_L) \tag{3.38}$$

and

$$\frac{\partial}{\partial \lambda_{(C_t,j,\vec{y}_L)}}[\frac{1}{Z_f}]\exp(\sum_{k=1}^{K}\lambda_{(c,k,\vec{y}_c)}I(x_k \in \vec{x}_\psi)I(\vec{y}_\psi = \vec{y}_c)) =$$

$$-\frac{1}{(Z_f)^2}\exp(\sum_{k=1}^{K}\lambda_{(c,k,\vec{y}_c)}I(x_k \in \vec{x}_\psi)I(\vec{y}_\psi = \vec{y}_c))$$

$$\frac{\partial}{\partial \lambda_{(C_t,j,\vec{y}_L)}}\sum_{\vec{y}_t}\exp(\sum_{k=1}^{K}\lambda_{(c,k,\vec{y}_t)}I(x_k \in \vec{x}_\psi)I(\vec{y}_\psi = \vec{y}_t)) = \tag{3.39}$$

$$-\frac{1}{(Z_f)^2}\exp(\sum_{k=1}^{K}\lambda_{(c,k,\vec{y}_c)}I(x_k \in \vec{x}_\psi)I(\vec{y}_\psi = \vec{y}_c))$$

$$\frac{\partial}{\partial \lambda_{(C_t,j,\vec{y}_L)}}\exp(\sum_{k=1}^{K}\lambda_{(c,k,\vec{y}_L)}I(x_k \in \vec{x}_\psi)I(\vec{y}_\psi = \vec{y}_L)) = \tag{3.40}$$

$$-Q(\vec{y}_\psi = \vec{y}_c)Q(\vec{y}_\psi = \vec{y}_L)I(x_j \in \vec{x}_\psi)I(\vec{y}_\psi = \vec{y}_L) \tag{3.41}$$

Plugging these into equation 3.36 we get

$$\sum_{f\in F}\sum_{\vec{y}_c}\frac{P^T(\vec{y}_c|x_f > 0)}{\hat{P}_\theta(\vec{y}_c|x_f > 0)}\frac{1}{Z_f}\sum_{\vec{x}\in U}\sum_{\psi\in C_f} I(x_f \in \vec{x}_\psi)I(x_j \in \vec{x}_\psi)I(\vec{y}_\psi = \vec{y}_L)$$
$$[Q(\vec{y}_\psi = \vec{y}_c) - Q(\vec{y}_\psi = \vec{y}_c)Q(\vec{y}_\psi = \vec{y}_L)] = \tag{3.42}$$

$$\sum_{f\in F}\sum_{\vec{y}_c}\frac{P^T(\vec{y}_c|x_f > 0)}{\hat{P}_\theta(\vec{y}_c|x_f > 0)}\frac{1}{Z_f}\sum_{\vec{x}\in U}\sum_{\psi\in C_f} I(x_f \in \vec{x}_\psi)I(x_j \in \vec{x}_\psi)I(\vec{y}_\psi = \vec{y}_L)$$
$$Q(\vec{y}_\psi = \vec{y}_c)[1 - Q(\vec{y}_\psi = \vec{y}_L)] \tag{3.43}$$

This does not involve any summation over the label space and can be tractably computed for any reasonable CRF, since the marginals of Q are easy to compute. This mean field approximation also allows the GE formulation to be extended to CRFs with even larger connectivity. Since we are no longer summing over the label space, the use of the GE regularizer is limited only by the number of factors in the model and tractability of calculating the mean field approximation.

3.4 Datasets

Three datasets were used to test the GE extention to CRFs. The Pascal 2010 dataset[12], the ICCV09 dataset [14], and the MSRC-v2 [1] dataset.

The ICCV09 and MSRC-v2 datasets have a ground truth, pixel level segmentation included, where as the Pascal dataset does not. For the Pascal data, segmentation was done using the Berkely Segmentation Engine[13]. As such, we expect the segmentation on the Pascal data to be much noisier than the other two datasets, and also more indicative of real world data.

Features are extracted from each segmentation region in the images, making up the unary features. For each region, the four closest regions (measured as the distance between the centers of mass) are defined as being adjacent. For the Pascal segmentation, there is also the notion of a child region, since the segmentation has a tree structure. A child region is a subsection of a larger parent region. Our set of binary factors is then over every pair of regions that are either adjacent, or have a parent-child relation. The binary features for these factor are the union of the unary features for the two regions.

To make the data easier to work with, and to reduce the time of training and inference, subsets where chosen to respresent a set number of classes from each data set. In the Pascal dataset, the classes car, motorcycle, person, bus, and background are present. In ICCV09 we have the classes mountain, tree, grass, road, and background. The MSRC data has six classes, namely building, grass, tree, sign, airplane, cow. The classes in each subset were chosen to give the largest number of images in the reduced set. A sixth class was added to MSRC to increase the subset to a more resonable size.

For the Pascal and ICCV09 datasets data is split into 5 images for training, 50 for validation, and 100 for testing. The training and validation sets are ensured to have at least 1 (for training) or 10 (for validation) images for each class. 245 images are left over in the unlabeled set for Pascal, and 272 for ICCV09.

MSRC has 6 images for training, again with at least one image for each class. Due to the smaller amount of data available, only 12 images are set aside for validation (2 for each class). This leaves 30 for testing and 39 for the unlabeled set.

[1] http://research.microsoft.com/en-us/projects/objectclassrecognition/

3.5 Experimental Methodology

We are interested in testing the hypothesis of whether labeled features can improve the classification of vision CRFs using a GE regularizer. As such, we will only be dealing with high quality labeled features. These labels are calculated using all-vs-one information gain on the unlabeled set as an oracle to identify the most informative features for each class. For unary feature labels, information gain is calculated over each region of the images, with none of the between region information being included. This can also be viewed as calculating information gain on the set of unary factors. Binary feature labels, which gives a pair of labels for features that appear in the union of two regions, are calculated over the set of binary factors. More specifically, we make a dataset where each instance represents two adjacent regions with features combined from both regions. The label for each instance is the concatenation of the labels from the regions. Information gain is calculated on this dataset to find the features that are most informative for the pair of labels. In all casese, I take the top 5 features with highest information gain for each label (or pairs of labels) to be in the labeled feature sets.

Our information gain is calculated as one-vs-all, for each class, and can include negative labels as well. That is, if the most informative feature for Class_1 is the absence of Feature_2, then we get the labeling of (Class_1, NOT_Feature_2). Negative labels are easy to incorporate into the GE formulation. Instead of doing indicator checks for the feature being present in a factor, we instead check if it is absent. The rest of the formulation is unchanged.

For the experiments performed in this paper, the datasets are split into training, testing, validation, and unlabeled sets, with 10 different splits for cross validation. Feature labels for each split are calculated using the unlabeled data. Training sets are kept small because GE has the best gains on small training sets. Care is taken to ensure that each class is still present in these small sets.

Experiments consist of running the baseline Conditional Random Field (CRF) against the CRF with GE regularizer for unary (GECRF_Unary), binary (GECRF_Binary), and unary and binary feature labels (GECRF) for each split of each dataset. The GE regularized CRFs are run with 3 random restarts, with the validation set used to chose the best of the 3. The experiments are then evaluated using the averaged Macro-F1 over each class. All CRF algorithms are trained using gradient ascent.

These experiments are repeated for two different inference algorithms, loopy belief propagation and mean field. These inference algorithms are used during the testing phase of the algorithm, and also during training to evaluate gradient directions. Loopy is a standard approximate inference algorithm used for CRFs, while the mean field inference algorithm better matches the assumptions of the GE regularizer term, which uses a mean field calculation in its derivation for both choices of inference algorithm. Comparing these two inference algorithms will illustrate how important it is for the assumptions of the GE term to match the assumptions of the likelihood term.

Chapter 4: Results

4.1 Experiment Results

In this section I report the outcome of the GECRF experiments. These experiments were done using the regular CRF as the baseline, with different levels of feature feedback for the GECRF algorithm. GECRF_Unary has labeled features for single regions. There are 5 labeled features for each class in this set. GECRF_Binary has labeled features for pairs of regions. There are 5 labeled features for each combination of labels in this set. Finally, GECRF uses all of the unary and binary features from the previous algorithms.

The experiments were done using two different approximate inference algorithms, loopy belief propagation and mean field. Loopy BP is the standard for CRFs with high connectivity, where as mean field better follows the assumptions of the GE regularizer formulation. The test is to see if following these assumptions improves the performance of using GE.

Results are reported using macro-averaged F1 scores. The F1 score for a single class is the formula $2 * \frac{precision*recall}{precision+recall}$. Averaging these values across all classes gives the macro-F1. This measure is chosen over regular accuracy because in the case of class imbalanced data accuracy can be misleading. In the Pascal and ICCV09 datasets, the background class has many more instances than the others. A classifier that just predicted background for every region could still get a relatively high accuracy. Macro-F1 takes into account both precision and recall, and averages the score between classes equally. Thus, it gives high score values to classifiers that can more reliably predict all classes.

Figure 4.1 has the results of using loopy belief propagation and Figure 4.2 has the results for mean field. Unfortunately, some of the GE results have not yet finished at this time. Of the finished results, we can see that the GECRF results for the ICCV09 dataset are significantly better than the CRF baseline for both inference algorithms. The MSRC GECRF results are only significantly better for the mean field inference algorithm. Baseline scores are quite poor in general. This is to be expected, since the

Loopy BP Results			
	Pascal	ICCV09	MSRC
CRF	0.166	0.148	0.350
GECRF_Unary	0.187	**0.209**	0.356
GECRF_Binary		**0.206**	0.352
GECRF		**0.189**	0.323

Figure 4.1: Macro-F1 results using loopy belief propogation. Results are averaged over 10 different training splits. Bold values are significantly different from the CRF baseline at the .05 level, using a paired value t-test.

Mean Field Results			
	Pascal	ICCV09	MSRC
CRF	0.171	0.138	0.138
GECRF_Unary	0.186	**0.209**	**0.204**
GECRF_Binary		**0.229**	
GECRF		**0.239**	

Figure 4.2: Macro-F1 results using mean field inference. Results are averaged over 10 different training splits. Bold values are significantly different from the CRF baseline at the .05 level, using a paired value t-test.

training set sizes are so small.

4.2 Timing Results

In this section I report the training time for each of the different algorithms. These are averaged over the 10 data splits used in the experiments. Because the GE algorithms use three random restarts during training, I only report the training time for the best restart, i.e. the one used for the F1 results in the previous section. On average, we can expect the training times for N random restarts to be N times longer. I do not report any timing results for classification because the GE regularizer only affects parameter learning and therefor does not change the classification time between the algorithms.

Figure 4.3 has the training time using loopy belief propagation and Figure 4.4 has the timing results using mean field. In both cases, the time to train is significantly increased

Loopy BP Timing			
	Pascal	ICCV09	MSRC
CRF	6.8 min	8.3 sec	15.6 min
GECRF_Unary	1.2 days	7.5 hr	2.3 hr
GECRF_Binary		21.9 hr	4.2 days
GECRF		14.0 hr	4.5 days

Figure 4.3: Training time using loopy belief propagation. Results are averaged over 10 training splits.

Mean Field Timing			
	Pascal	ICCV09	MSRC
CRF	37 min	24.7 sec	1.2 hr
GECRF_Unary	3.7 days	4.6 hr	2.7 days
GECRF_Binary		17.3 hr	
GECRF		19.0hr	

Figure 4.4: Training time using mean field inference. Results are averaged over 10 training splits.

when labeled features are included. The reasons and ramifications of this are touched on in section 5.1.

4.3 Learning Curves

In the experiments illustrated in the previous sections, we saw how the baseline CRF algorithm with a small amount of training data can be improved through the addition of labeled features and a set of unlabeled images. This begs the question, how do these improvements compare to those gained from just adding more training data?

To get an idea of this, we can compare the GECRF results to a CRF trained with incrementally increasing training data. This is done for the ICCV09 data in figure 4.5. Here we are looking at the best baseline CRF (trained with loopy BP) compared to the best GECRF algorithm (mean field inference algorithm, with unary and binary features). The results indicate that we can expect roughly the same improvement by

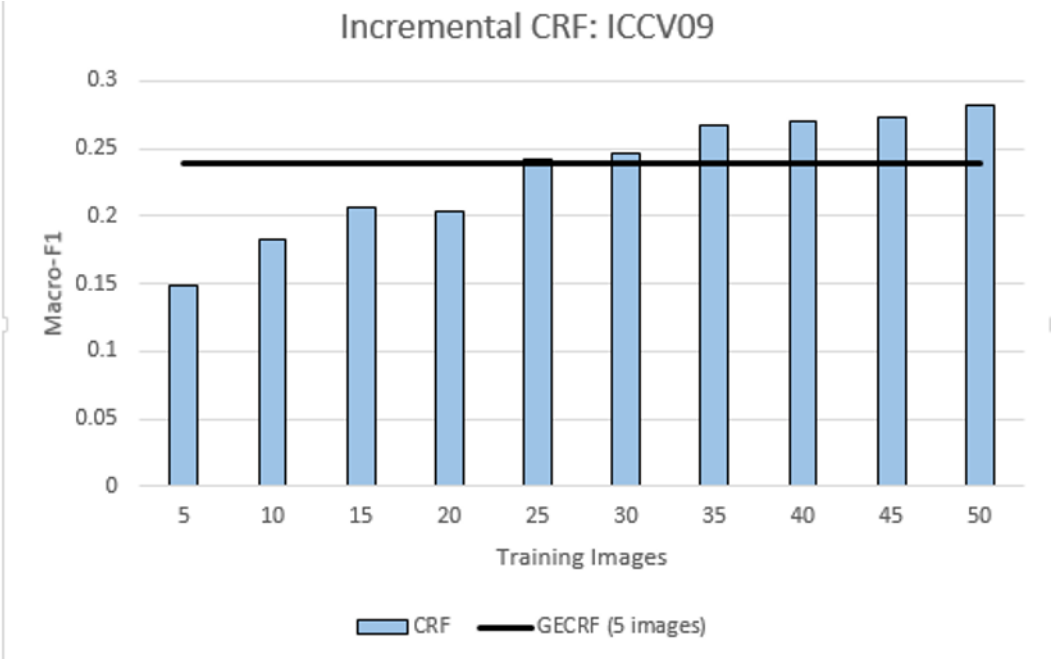


Figure 4.5: Comparison of an incrementally trained CRF to GECRF for ICCV09. The CRF is trained using loopy BP with an increasing number of training examples. GECRF is trained with mean field and includes unary and binary features. Results are averaged over 10 training splits.

adding 20 labeled images to the training set. Keep in mind however that labeling a single image involves labeling all the regions in the image. In the ICCV09 dataset there are approximately 20 regions per image, so this would be an additional 400 labelings in total. This is compared to the 150 features labeled in the GECRF algorithm.

Chapter 5: Discussion

In the experiment results, we can see that the improvement over the baseline CRF algorithm is greater for the mean field inference algorithm than for loopy belief propagation. The improvement using loopy BP is only significant for the ICCV09 dataset, and essentially zero for MSRC. On the other hand, mean field shows substantial improvement for ICCV09 and MSRC, with negligable improvement on the Pascal dataset.

As stated in section 3.5, the purpose of testing these two inference algorithms was to test the importance of matching assumptions of the GE regularizer term. The GE regularizer uses a mean field assumption when calculating the gradient for gradient ascent, so using mean field inference would ensure that the likelihood term agrees with the regularizer. The results indicate that this agreement is important to maintain, as the loopy BP inference algorithm struggled to improve over the baseline CRF.

In terms of absolute performance, using loopy BP produced a more accurate baseline CRF. Changing the mean field approximation in the GE formulation to one that better matches loopy BP, or some other more accurate inference algorithm, could bolster the algorithm by giving it a better starting point. How much effect this might have is hard to tell though, as the the mean field results for ICCV09 did eventually surpass the loopy results with the addition of more features. This would also involve re-deriving the approximation for GE to match the inference algorithm, and is left as a subject for future work.

5.1 Training Time of GECRF

The substantial increase in training time for the GECRF algorithm is an obvious issue that needs to be resolved. The training bottleneck comes from having to evaluate the KL divergence, which requires calculating the empirical distribution over the entire unlabeled dataset. As illustrated in equation 3.43, calculating the gradient of the regularizer involves performing inference on ever factor in ever image in the unlabeled set. This calculation is done as an inner loop for gradient ascent, causing the large increase in

training time. This increase will only get worse as more unlabeled instances are added to the pool, which is another incentive for finding a solution. Unlabeled data is cheap, and we would like to be able to add a large amount of it to the algorithm without having to pay such a steep time penalty.

One possible solution to this could be using parallelization. The inference for each image in the unlabeled set is done independently of the other images, so this calculation could be optimized by distributing the unlabeled data over multiple processors. On average, this should give a speed up factor equal to the number of processors used. Because this bottleneck accounts for the majority of training time, the overall speed up should be similar, particularly when there is a large amount of unlabeled data being used. As a proof of concept, I ran one split of the parallelized code for MSRC with unary features and loopy BP inference algorithm on a machine with 8 processors. The algorithm took 47.2 minutes to train, compared to the average training time of 2.3 hours. We would expect this speed up to be even greater for the other two data sets, which have much larger pools of unlabeled data.

Another simple solution would be to abondon the gradient ascent approach and use a different algorithm for parameter learning. There have been many methods proposed for speeding up CRF parameter learning [37, 18], any of which could help the GECRF run in a more feasible time frame. The choice of gradient ascent was done for convenience as it is a common out of the box approach, and there are many more options that could be tried for the sake of fast parameter learning. It is also worth mentioning that the GE regularizer does not affect classification time, so any trained algorithm will perform as fast as the CRF baseline once deployed.

5.2 Incorporating User Feedback

The work in this paper shows that oracle labeled features can improve image classification with the right inference algorithm. The next step is to test the effect of feedback from real users. Unlike the text domain that GE has been used in previously, having users label features for images would likely not yield useful results. This is because the features used for image classication (particularly the ones used in this implementation) are extremely unintuitive. Most users are not going to be able to specify the SIFT vector that best identifies cars. In order to use human feedback, the labeling is going to have to be on

attributes, not features.

Attributes are higher level qualities of objects that are human interpretable, such as the concepts of "soft", "metalic", "wet", etc. This is the level of feedback used in [3], and usually requires an identifier for each attribute, to classify each object as having the property or not. The GE framework is easily extendable to work at the level of attributes instead of features. This is because the empirical distribution is defined as the conditional distribution where the labeled feature (or attribute) is present. So long as the labeled quality is detectable for an image region, it can be an input to GE. This opens the door to users labeling attributes, combinations of attributes, relational properties of objects, or any other image quality that can be automatically detected.

It would be useful to test the benefits of using the GECRF in an active learning scenario, similar to [3] and [11], where the machine prompts the users for labels on attributes. In the text experiments, Druck et al found that having users label features rather than instances produced better classification algorithms given the same amount of annotation time. It is possible that this may be the case with providing attribute labels for images as well.

Chapter 6: Conclusion

In this paper I derived a new version of generalized expectation to incorporate labeled features into image classification. Experiments were run with oracle feature labelings to provide a proof of concept for the potential benefit of gathering this feedback from users. Results indicate that, with the correct choice of inference algorithm, the labeled feature feedback improves classication over the baseline algorithm. With further improvements made to optimize the training time, the algorithm could be easily extended to solict attribute feedback from users in an active learning environment. Previous experiments in feature labeling for text documents showed that soliciting labels for features gives better results for the same amount of time spent labeling instances. The preliminary work done here supports the claim that this could be the case for image classification as well, and encourages extending this work to include human feedback in the future.

Bibliography

[1] S. Bandla and K. Grauman. Active learning of an action detector from untrimmed videos. In *IEEE International Conference on Computer Vision (ICCV), Sydney*. IEEE, 2013.

[2] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *NIPS*, volume 2, page 3, 2000.

[3] Arijit Biswas and Devi Parikh. Simultaneous active learning of classifiers & attributes via relative feedback. *CVPR*, 2013.

[4] Luca Cagliero and Paolo Garza. Improving classification models with taxonomy information. *Data & Knowledge Engineering*, 2013.

[5] Gustavo Camps-Valls, T Bandos Marsheva, and Dengyong Zhou. Semi-supervised graph-based hyperspectral image classification. *Geoscience and Remote Sensing, IEEE Transactions on*, 45(10):3044–3054, 2007.

[6] Ming-Wei Chang, Lev Ratinov, and Dan Roth. Guiding semi-supervision with constraint-driven learning. *Urbana*, 51:61801, 2007.

[7] Ming-Wei Chang, Lev Ratinov, and Dan Roth. Structured learning with constrained conditional models. *Machine learning*, 88(3):399–431, 2012.

[8] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[9] Shubhomoy Das, Travis Moore, Weng-Keen Wong, Simone Stumpf, Ian Oberst, Kevin McIntosh, and Margaret Burnett. End-user feature labeling: Supervised and semi-supervised approaches based on locally-weighted logistic regression. *Artificial Intelligence*, 2013.

[10] Gregory Druck, Gideon Mann, and Andrew McCallum. Learning from labeled features using generalized expectation criteria. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 595–602. ACM, 2008.

[11] Gregory Druck, Burr Settles, and Andrew McCallum. Active learning by labeling features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 81–90. Association for Computational Linguistics, 2009.

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[13] Charless Fowlkes, David Martin, and Jitendra Malik. Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–54. IEEE, 2003.

[14] Stephen Gould, Richard Fulton, and Daphne Koller. Decomposing a scene into geometric and semantically consistent regions. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1–8. IEEE, 2009.

[15] A Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Scalable active learning for multi-class image classification. *IEEE*, 2012.

[16] Ben King and Steven Abney. Labeling the languages of words in mixed-language documents using weakly supervised methods. In *Proceedings of NAACL-HLT*, pages 1110–1119, 2013.

[17] Adriana Kovashka and Kristen Grauman. Attribute pivots for guiding relevance feedback in image search. ICCV, 2013.

[18] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *arXiv preprint arXiv:1210.5644*, 2012.

[19] Jing Li and Nigel M Allinson. Relevance feedback in content-based image retrieval: A survey. In *Handbook on Neural Information Processing*, pages 433–469. Springer, 2013.

[20] Xin Li and Yuhong Guo. Adaptive active learning for image classification. CVPR, 2013.

[21] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[22] Gideon S Mann and Andrew McCallum. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *ACL*, volume 8, pages 870–878, 2008.

[23] Andrew McCallum, Gideon Mann, and Gregory Druck. Generalized expectation criteria. *Computer science technical note, University of Massachusetts, Amherst, MA*, 2007.

[24] Roozbeh Mottaghi, Sanja Fidler, Jian Yao, Raquel Urtasun, and Devi Parikh. Analyzing semantic segmentation using hybrid human-machine crfs. *CVPR*, 2013.

[25] Haiwei Pan, Niu Zhang, Qilong Han, and Guisheng Yin. Incorporating domain knowledge into multistrategical image classification. In *Database Technology and Applications (DBTA), 2010 2nd International Workshop on*, pages 1–4. IEEE, 2010.

[26] Amar Parkash and Devi Parikh. Attributes for classifier feedback. In *Computer Vision–ECCV 2012*, pages 354–368. Springer, 2012.

[27] Hema Raghavan and James Allan. An interactive algorithm for asking and incorporating feature feedback into support vector machines. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 79–86. ACM, 2007.

[28] David Rim, Kamrul Hassan, and Christopher J Pal. Semi supervised learning for wild faces and video. In *BMVC*, pages 1–12, 2011.

[29] Robert E Schapire, Marie Rochery, Mazin Rahim, and Narendra Gupta. Incorporating prior knowledge into boosting. In *ICML*, volume 2, pages 538–545, 2002.

[30] Umamahesh Srinivas, Yuanming Suo, Minh Dao, Vishal Monga, and Trac D Tran. Structured sparse priors for image classification. *ICIP*, 2013.

[31] André Stumpf, Nicolas Lachiche, J-P Malet, Norman Kerle, and Anne Puissant. Active learning in the spatial domain for remote sensing image classification. *IEEE*, 2013.

[32] Michael J Swain and Dana H Ballard. Indexing via color histograms. In *Active Perception and Robot Vision*, pages 261–273. Springer, 1992.

[33] Roberto Tronci, Gabriele Murgia, Maurizio Pili, Luca Piras, and Giorgio Giacinto. Imagehunter: a novel tool for relevance feedback in content based image retrieval. In *New Challenges in Distributed Information Filtering and Retrieval*, pages 53–70. Springer, 2013.

[34] Devis Tuia, Frédéric Ratle, Fabio Pacifici, Mikhail F Kanevski, and William J Emery. Active learning methods for remote sensing image classification. *Geoscience and Remote Sensing, IEEE Transactions on*, 47(7):2218–2232, 2009.

[35] Sudheendra Vijayanarasimhan and Kristen Grauman. Multi-level active prediction of useful image annotations for recognition. In *Advances in Neural Information Processing Systems*, pages 1705–1712, 2008.

[36] Sudheendra Vijayanarasimhan and Kristen Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1449–1456. IEEE, 2011.

[37] SVN Vishwanathan, Nicol N Schraudolph, Mark W Schmidt, and Kevin P Murphy. Accelerated training of conditional random fields with stochastic gradient methods. In *Proceedings of the 23rd international conference on Machine learning*, pages 969–976. ACM, 2006.

[38] Mengqiu Wang and Christopher D Manning. Cross-lingual pseudo-projected expectation regularization for weakly supervised learning. *arXiv preprint arXiv:1310.1597*, 2013.