Chapter 4
Discriminative Graphical Models for Context-Based Classification

Sanjiv Kumar

Abstract. Natural image data shows significant dependencies that should be modeled appropriately to achieve good classification. Such dependencies are commonly referred to as *context* in Vision. This chapter describes Conditional Random Fields (CRFs) based discriminative models for incorporating context in a principled manner. Unlike the traditional generative Markov Random Fields (MRFs), CRFs allow the use of arbitrarily complex dependencies in the observed data along with data-dependent interactions in labels. Fast and robust parameter learning techniques for such models are described. The extensions of the standard binary CRFs to handle problems with multiclass labels or hierarchical context are also discussed. Finally, application of CRFs on contextual object detection, scene segmentation and texture recognition tasks is demonstrated.

4.1 Contextual Dependencies in Images

One of the fundamental problems in computer vision is that of *image understanding* or *semantic scene interpretation* i.e., to interpret the scene contained in an image as a collection of meaningful entities. This may involve parsing information in the scene at different levels. Here, we focus on the problem of classification or labeling of various components in natural images, where a component may be an image pixel, a region, an object or the entire image itself.

The problem of detecting and classifying regions and objects in images is a challenging task due to ambiguities in the appearance of the visual data. The use of context can help alleviate this problem significantly. For example, as shown in Figure 4.1, just on the basis of appearance, it may be difficult to differentiate a sky patch from a water patch but their relative spatial configuration with respect to other regions removes this ambiguity. Similarly, a patch from a tree may appear locally very

Sanjiv Kumar
Google Research, New York, USA
e-mail: sanjivk@google.com

R. Cipolla, S. Battiato, & G.M. Farinella (Eds.): Computer Vision, SCI 285, pp. 109–134.
springerlink.com

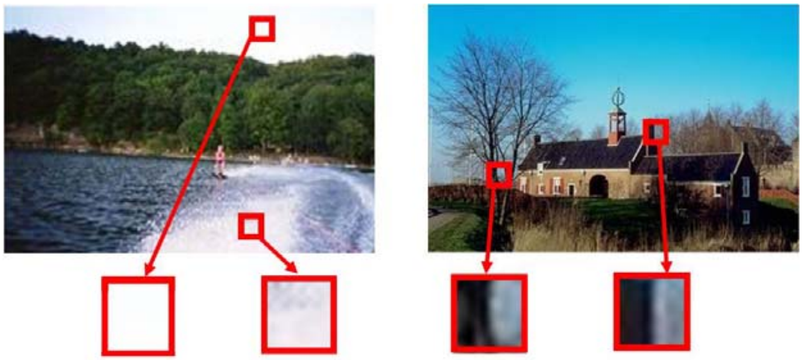


Fig. 4.1 Classification of image components is difficult due to ambiguities in their appearance. In the left image, sky and water regions look similar while in the right image, tree and building regions look similar. Context can help resolve these ambiguities.

similar to another patch from a building (Figure 4.1, right image). But if we look at larger neighborhoods of the patch, it is easy to classify which patch is a building patch.

It is well known that natural images are not a random collection of independent pixels. The spatial arrangement of pixels (or blocks) in images is crucial to make a meaningful image. It is important to use contextual information in the form of spatial dependencies for robust analysis of images. Since these dependencies can be short-range or long-range, one would like to have total freedom in modeling data interactions without restricting oneself to small local neighborhoods. This idea forms the core of the work described in this chapter. The spatial dependencies may vary from being local to global and the challenge is how to maintain global spatial consistency using models that only need to consider relatively local dependencies.

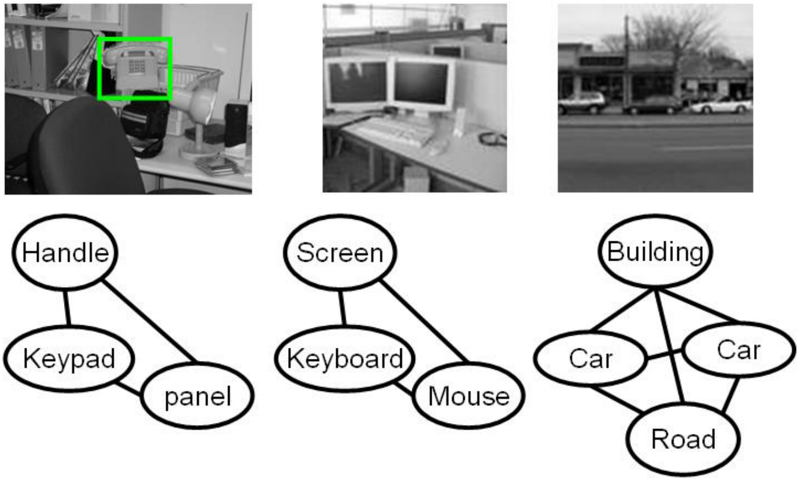
4.1.1 The Nature of Contextual Interactions

There are several types of contextual interactions one would like to model to achieve robust classification in images. The simplest type of interaction is based on the notion of spatial smoothness of labels in natural images. According to this, neighboring pixels tend to have similar labels (except at the discontinuities). For example, if a pixel in left image in Figure 4.1 has label *sky*, there is a high probability that the neighboring pixels also have the same label except at the boundaries. In fact, the underlying smoothness of natural images forms the basis for recovering the true image from its noisy version in image denoising applications. These type of interactions are generally restricted to the pixel level. However, in addition to these, there exist significant interactions among bigger regions in images. In the previous example (Figure 4.1, left image), different semantic regions follow plausible spatial configurations (e.g., sky tends to occur above water or vegetation).

In addition to the interaction in labels, there are also complex interactions in the observed data that might be required for classification purposes. Consider the task of detecting structured textures (e.g., man-made structures such as buildings) in a given image. The data belonging to this type of textures is highly dependent on its neighbors. This is because, in man-made structures, the lines or edges at spatially adjoining regions follow some underlying organization rules rather than being random (see Figure 4.1, right image).

Now, considering the case of parts-based object detection, one would like to detect different parts of an object to form a hypothesis about the presence of the whole object. For example, in Figure 4.2 (a), we are interested in detecting a *phone*. Different parts of the phone such as handle, keypad and front panel are related to each other through geometric and, possibly, photometric constraints. The phone can be detected in the scene if we can find the locations of these parts. However, to reliably detect these parts, we need to encode not only the appearance of each individual part but also the spatial relationships among the parts. Thus, in this case, context is applied using the mutual relationships of different parts.

Finally, the contextual interactions for object detection are not limited to the parts of a single object. These may include interactions among various objects or regions in the scene. For example, as shown in Figure 4.2 (b), the presence of a monitor screen increases the probability of having a keyboard or mouse nearby. Exploiting such contextual information is crucial especially for detecting those objects



(a) (b) (c)

Fig. 4.2 Context is important for the detection of objects in their natural surroundings. (a) Different parts of an object (phone) are related through geometric constraints that can help in robust detection of individual parts. (b) Different objects (monitor, keyboard and mouse) in a scene occur in restricted configurations which can help in detecting objects with impoverished appearance (e.g., mouse). (c) Context from other regions (e.g., buildings and roads can be helpful in detecting cars).

that have impoverished appearances such as the mouse in this case. Similarly, the presence of regions such as buildings and roads in a scene restricts the possible locations a car can take in the image (Figure 4.2 (c)).

To summarize, context in images can be broadly divided into two categories. First, *local context* e.g., local smoothness of pixel labels in images or interactions among different parts of an object, and second, *global context* such as interaction among bigger objects and regions in images. The challenge is how to model different types of context, which may include complex dependencies in the observed image data as well as the labels, in a principled manner. Ideally, one would like to find a computational model that can learn all relevant types of context automatically in a single consistent framework using the training data. Discriminative graphical models provide a solid platform to achieve that. Such models are by nature noncausal and are typically represented by undirected graphs. Let us first briefly review an undirected probabilistic graphical model commonly used in Computer Vision.

4.2 Markov Random Field (MRF)

Markov Random Fields (MRFs) are the most popular undirected graphical models in vision, which allow one to incorporate local contextual constraints in labeling problems in a principled manner. MRFs were made popular in vision by early work of Geman and Geman [4], and Besag [1]. MRFs are generally used in a probabilistic generative framework that models the joint probability of the observed data and the corresponding labels. In other words, let y be the observed data from an input image, where $y = \{y_i\}_{i \in S}$, y_i is the data from the i^{th} site, and S is the set of sites. Let the corresponding labels at the image sites be given by $x = \{x_i\}_{i \in S}$. In the MRF framework, the posterior over the labels given the data is expressed using the Bayes' rule as,

$$P(x|y) \propto p(x,y) = P(x)p(y|x)$$

where the prior over labels, $P(x)$ is modeled as a MRF. For computational tractability, the observation or likelihood model, $p(y|x)$ is assumed to have a factorized form, i.e., $p(y|x) = \prod_{i \in S} p(y_i|x_i)$. However, this assumption is too restrictive for several natural image analysis applications. For example, consider a class that contains man-made structures (e.g., buildings). The data belonging to such a class is highly dependent on its neighbors. This is because, in man-made structures, the lines or edges at spatially adjoining sites follow some underlying organization rules rather than being random. This is also true for a large number of texture classes that are made of structured patterns.

Another thing to note is that the interaction among labels in MRFs is modeled by the term $P(x)$, which is seen as a prior in the Bayesian view. The main drawback of this view is that the label interactions do not depend on the observed data y. This prohibits one from modeling data-dependent interactions in labels that are necessary for a variety of tasks. For example, while implementing local smoothness of labels in image segmentation, it may be desirable to use observed data to modulate the smoothness according to the image intensity gradients. Further, in parts based object detection, to model interactions among object parts, we need observed data to enforce geometric (and possibly photometric) constraints. This is also the case for modeling higher level interactions between objects or regions in an image. As we will see later, discriminative graphical models allow interactions among labels based on unrestricted use of observations as necessary. This step is crucial to develop models that can incorporate interactions of different types within the same framework.

In MRF formulation of binary classification problems, the label interaction field $P(x)$ is commonly assumed to be a homogeneous and isotropic Ising model (or Potts model for multiclass labeling problems) with only pairwise nonzero potentials. If the data likelihood $p(y|x)$ is approximated by assuming that the observed data is conditionally independent given the labels, the posterior distribution[1] over labels can be written as,

$$P(x|y) = \frac{1}{Z_m} \exp\left(\sum_{i \in S} \log p(s_i(y_i)|x_i) + \sum_{i \in S} \sum_{j \in \mathcal{N}_i} \beta_m x_i x_j \right), \tag{4.1}$$

where β_m is the interaction parameter of the MRF, and $s_i(y_i)$ is a *single-site* feature vector, which uses data only from a single site i, i.e., y_i. Note that even though only the label prior, $P(x)$ was assumed to be a MRF, the assumption of conditional independence of data implies that the posterior given in (4.1) is also a MRF. This allows one to reap the benefits of readily available tools of inference over a MRF. If the conditional independence assumption is not used, the posterior will usually not be a MRF making the inference difficult.

Now, if we turn our attention again toward the original aim, we are interested in classification of image sites. For classification purposes, we want to estimate the posterior over labels given the observations, i.e., $P(x|y)$. In a generative framework, one expends efforts to model the joint distribution $p(x,y)$, which involves implicit modeling of the observations. In a discriminative framework, one models the distribution $P(x|y)$ directly. A major advantage of doing this is that the true underlying generative model may be quite complex even though the class posterior is simple. This means that the generative approach may spend a lot of resources on modeling the generative models which are not particularly relevant to the task of inferring the class labels. Moreover, learning the class density models may become even harder when the training data is limited. The discriminative approach saves one from making simplistic assumptions about the data. This view forms the core theme of the model discussed in the following Sections.

4.3 Conditional Random Field (CRF)

Conditional Random Fields (CRFs) were originally proposed by Lafferty et al. [15] in the context of segmentation and labeling of 1-D text sequences. CRFs are discriminative models that directly model the conditional distribution over labels

[1] With a slight abuse of notation, we will use the term 'MRF model' to indicate this posterior.

i.e., $P(x|y)$ as a Markov Random Field. This approach allows one to capture arbitrary dependencies between the observations without resorting to any model approximations. In this chapter, we will follow the generalized version of CRFs proposed by Kumar and Hebert [11] and [12]. They first introduced the extension of original 1-D CRFs to 2-D graphs over images. Their version also allows the use of arbitrary discriminative classifiers to model different types of interactions in labels and data, leading to more flexible and powerful generalization of CRFs.

We first restate the definition of CRFs as given by Lafferty et al. [15]. Let the observed data from an input image be given by $y = \{y_i\}_{i \in S}$ where y_i is the data from i^{th} site and $y_i \in \Re^c$. The corresponding labels at the image sites are given by $x = \{x_i\}_{i \in S}$. First let us focus on binary classification problem, i.e. $x_i \in \{-1, 1\}$. Section 4.5.1 will describe its extension to multiclass labeling problem. The random variables x and y are jointly distributed, but in a discriminative framework, a conditional model $P(x|y)$ is constructed from the observations and labels, and the marginal $p(y)$ is not modeled explicitly.

Definition 4.1. CRF: 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: $P(x_i|y, x_{S-\{i\}}) = P(x_i|y, x_{\mathcal{N}_i})$, where $S - \{i\}$ is the set of all the nodes in the graph except the node i, $\mathcal{N}_i$ is the set of neighbors of the node i in G, and x_Ω represents the set of labels at the nodes in set Ω.

Thus, a CRF is a random field globally conditioned on the observations y. The condition of positivity requiring $P(x|y) > 0$, $\forall$ x has been assumed implicitly. Using the Hammersley-Clifford theorem [6] and assuming only up to pairwise clique potentials to be nonzero, the conditional distribution over all the labels x given the observations y in a CRF can be written as,

$$P(x|y) = \frac{1}{Z} \exp\left(\sum_{i \in S} A_i(x_i, y) + \sum_{i \in S} \sum_{j \in \mathcal{N}_i} I_{ij}(x_i, x_j, y) \right), \tag{4.2}$$

where Z is a normalizing constant known as the partition function, and -A_i and -I_{ij} are the unary and pairwise potentials respectively. With a slight abuse of notation, we will call A_i the *association potential* and I_{ij} the *interaction potential*.

There are two main differences between the conditional model given in Equation (4.2) and the traditional MRF framework given in Equation (4.1). First, in the conditional fields, the association potential at any site is a function of all the observations y while in MRFs (with the assumption of conditional independence of the data), the association potential is a function of data only at that site, i.e., y_i. Second, the interaction potential for each pair of nodes in MRFs is a function of only labels, while in the conditional models it is a function of labels as well as all the observations y. As will be shown later, these differences play a crucial role in modeling arbitrary interactions in both observed data and labels in natural images in a principled manner.

In this discussion, we assume the random field given in Equation (4.2) to be homogeneous, i.e., the functional forms of A_i and I_{ij} are independent of the location i.

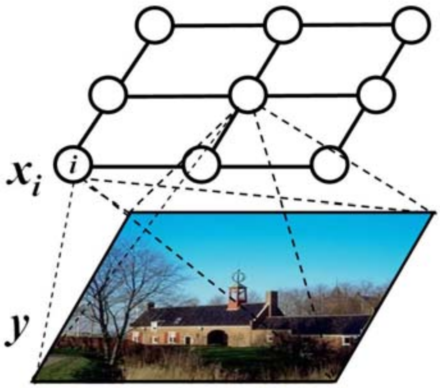


Fig. 4.3 An illustration of a typical CRF for an example task of man-made structure detection in natural images. The aim is to label each site i.e., each 16 × 16 image block whether it is a man-made structure or not. The top layer represents the labels on all the image sites. Note that each site i can potentially use features from the whole image y unlike the traditional MRFs.

In addition, we also assume the field to be isotropic implying that the label interactions are non-directional. In other words, I_{ij} is independent of the relative locations of sites i and j. Thus, subsequently we will drop the subscripts and simply use the notation A and I to denote the two potentials. In fact, the assumption of isotropy can be easily relaxed at the cost of a few additional parameters. Thus, we will consider models of the following form:

$$P(x|y) = \frac{1}{Z}\exp\left(\sum_{i\in S} A(x_i,y) + \sum_{i\in S}\sum_{j\in\mathcal{N}_i} I(x_i,x_j,y)\right). \tag{4.3}$$

Due to this form of CRFs, it is possible to treat different applications from low-level image denoising to high-level contextual object detection seamlessly in a single framework. Figure 4.3 illustrates a typical CRF for an example image analysis task of man-made structure detection. Suppose, we are given an input image y shown in the bottom layer and we are interested in labeling each image site (in this case a 16 × 16 image block) whether it contains a man-made structure or not. The top layer represents the labels x on all the image sites. Note that each site i can potentially use features from the whole image y unlike the traditional MRFs. In addition, CRFs allow to use image data to model interactions between two neighboring sites i and j. The following sections describe how the unary and the pairwise potentials are designed in CRFs.

4.3.1 Association Potential

In the CRF framework, the association potential, $A(x_i,y)$, can be seen as a measure of how likely a site i will take label x_i given image y, ignoring the effects of other sites in the image (Figure 4.4). Suppose, $f(.)$ is a function that maps an arbitrary patch in an image to a feature vector such that $f : \mathcal{Y}_p \rightarrow \Re^l$. Here $\mathcal{Y}_p$ is the set of all possible patches in all possible images. Let $\omega_i(y)$ be an arbitrary patch in the neighborhood of site i in image y from which we want to extract a feature vector $f(\omega_i(y))$. Note that the neighborhood used for the patch $\omega_i(y)$ need not be the same

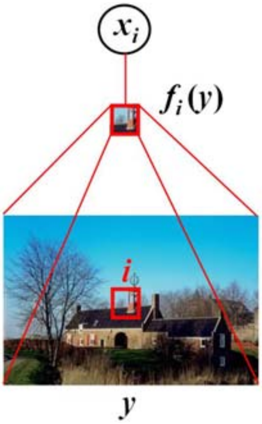


Fig. 4.4 Given a feature vector $f_i(y)$ at site i, the association potential in CRFs can be seen as a measure of how likely the site i will take label x_i, ignoring the effects of other sites in the image. Note that the feature vector $f_i(y)$ can be constructed by pooling arbitrarily complex dependencies in the observed data y.

as the label neighborhood $\mathcal{N}_i$. Indeed, $\omega_i(y)$ can potentially be the whole image itself. For clarity, let us denote the feature vector $f(\omega_i(y))$ at each site i by $f_i(y)$. The subscript i indicates the difference just in the feature vectors at different sites, *not* in the functional form of $f(.)$. Then, $A(x_i, y)$ is modeled using a local discriminative model that outputs the association of the site i with class x_i as,

$$A(x_i, y) = \log P'(x_i | f_i(y)), \tag{4.4}$$

where $P'(x_i|f_i(y))$ is the local class conditional at site i. This form allows one to use an arbitrary domain-specific probabilistic discriminative classifier for a given task. This can be seen as a parallel to the traditional MRF models where one can use arbitrary local generative classifier to model the unary potential. One possible choice of $P'(.)$ is Generalized Linear Models (GLM), which are used extensively in statistics to model the class posteriors [18]. Logistic function is a commonly used link in GLMs although other choices such as probit link exist. Using a logistic function, the local class conditional can be written as,

$$P'(x_i{=}1|f_i(y)) = \frac{1}{1+e^{-(w_0+w_1^T f_i(y))}} = \sigma(w_0 + w_1^T f_i(y)), \tag{4.5}$$

where $w = \{w_0, w_1\}$ are the model parameters. This form of $P'(.)$ will yield a linear decision boundary in the feature space spanned by vectors $f_i(y)$. To extend the logistic model to induce a nonlinear decision boundary, a transformed feature vector at each site i can be defined as $h_i(y) = [1, \phi_1(f_i(y)), \ldots, \phi_R(f_i(y))]^T$ where $\phi_k(.)$ are arbitrary nonlinear functions. These functions can be seen as explicit kernel mapping of the original feature vector into a high dimensional space. The first element of the transformed vector is kept as 1 to accommodate the bias parameter w_0. Further, since $x_i \in \{-1, 1\}$, the probability in Equation (4.5) can be compactly expressed as,

$$P'(x_i|y) = \sigma(x_i w^T h_i(y)). \tag{4.6}$$

Finally, for this choice of $P'(.)$, the association potential can be written as,

$$A(x_i,y) = \log(\sigma(x_i w^T h_i(y))) \tag{4.7}$$

This transformation ensures that the CRF is equivalent to a logistic classifier if the interaction potential in Equation (4.3) is set to zero. Besides GLMs, discriminative classifiers based on SVM, Neural Network and Boosting have been successfully used in modeling association potential in the literature. Note that in Equation (4.7), the transformed feature vector at *each* site i i.e., $h_i(y)$ is a function of the whole set of observations y. This allows one to pool arbitrarily complex dependencies in the observed data for the purpose of classification. On the contrary, the assumption of conditional independence of the data in the traditional MRF framework allows one to use data only from a particular site, i.e., y_i, to design the log-density, which acts as the association potential as shown in Equation (4.1).

4.3.2 Interaction Potential

In the CRF framework, the interaction potential can be seen as a measure of how the labels at neighboring sites i and j interact given the observed image y (Figure 4.5). To model the interaction potential, $I(.)$, we first analyze a form commonly used in the MRF framework. For the isotropic, homogeneous Ising model, the interaction potential is given as $I(.) = \beta x_i x_j$, which penalizes every dissimilar pair of labels by the cost β [8]. This form of interaction favors piecewise constant smoothing of the labels without considering the discontinuities in the observed data explicitly. Geman and Geman extended the Ising model to a line-process model which allows discontinuities in labels through piecewise continuous smoothing [4].

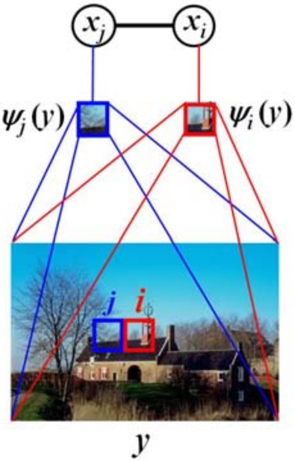


Fig. 4.5 Given feature vectors $\psi_i(y)$ and $\psi_j(y)$ at two neighboring sites i and j respectively, the interaction potential can be seen as a measure of how the labels at sites i and j influence each other. Note that such interaction in labels is dependent on the observed image data y, unlike the traditional generative MRFs.

However, such discontinuity adaptive models also do not use the observed data to model the discontinuities.

In contrast, in the CRF formulation, the interaction potential is a function of all the observations y. Suppose, $\psi(.)$ is a function that maps an arbitrary patch in an image to a feature vector such that $\psi : \mathcal{Y}_p \rightarrow \Re^{\gamma}$. Let $\Omega_i(y)$ be an arbitrary patch in the neighborhood of site i in image y from which we want to extract a feature vector $\psi(\Omega_i(y))$. Note that the neighborhood used for the patch $\Omega_i(y)$ need not be the same as the label neighborhood $\mathcal{N}_i$. For clarity, let us denote the feature vector $\psi(\Omega_i(y))$ at each site i by $\psi_i(y)$. Similarly, we define a feature vector $\psi_j(y)$ for site j. Again, to emphasize, the subscripts i and j indicate the difference just in the feature vectors at different sites, *not* in the functional form of $\psi(.)$. Given the features at two different sites, we want to learn a pairwise discriminative model $P''(x_i{=}x_j|\,\psi_i(y), \psi_j(y))$. Note that by choosing the function ψ_i to be different from f_i, used in Equation (4.5), information different from f_i can be used to model the relations between pairs of sites.

For a pair of sites (i, j), let $\mu_{ij}(\psi_i(y), \psi_j(y))$ be a new feature vector such that $\mu_{ij} : \Re^{\gamma} \times \Re^{\gamma} \rightarrow \Re^{q}$. Denoting this feature vector as $\mu_{ij}(y)$ for simplification, the interaction potential is modeled as,

$$I(x_i, x_j, y) = x_i x_j v^T \mu_{ij}(y), \tag{4.8}$$

where v are the model parameters. Note that the first component of $\mu_{ij}(y)$ is fixed to be 1 to accommodate the bias parameter. There are two interesting properties of the interaction potential given in Equation (4.8). First, if the association potential at each site and the interaction potentials for all the pairwise cliques except the pair (i, j) are set to zero in Equation (4.3), the CRF acts as a logistic classifier which yields the probability of the site pair to have the same labels given the observed data. Of course, one can generalize the form in Equation (4.8) as,

$$I(x_i, x_j, y) = \log P''(x_i, x_j|\,\psi_i(y), \psi_j(y)), \tag{4.9}$$

similar to the association potential defined in Section 4.3.1 and can use arbitrary pairwise discriminative classifier to define this term. The second property of the interaction potential form given in Equation (4.8) is that it generalizes the Ising model. The original Ising form is recovered if all the components of vector v other than the bias parameter are set to zero in Equation (4.8). A geometric interpretation of interaction potential is that it partitions the space induced by the relational features $\mu_{ij}(y)$ between the pairs that have the same labels and the ones that have different labels. Hence Equation (4.8) acts as a data-dependent discontinuity adaptive model that will moderate smoothing when the data from the two sites is 'different'. The data-dependent smoothing can especially be useful to absorb the errors in modeling the association potential. Anisotropy can be easily included in the CRF model by parameterizing the interaction potentials of different directional pairwise cliques with different sets of parameters v.

4.4 Parameter Learning and Inference

One of the crucial requirements to make the CRF-based models applicable to a variety of real-world tasks is accurate and efficient parameter learning in these models. Here, we focus on maximum likelihood based supervised learning of CRFs.

For 1-D sequential CRFs proposed by Lafferty et al. [15], exact maximum likelihood parameter learning is feasible because the induced graph does not contain loops. However, when a graph contains loops, it is generally infeasible to exactly maximize the likelihood with respect to the parameters. Therefore, a critical issue in applying CRFs to image-based applications is the design of effective parameter learning techniques that can operate on arbitrary graphs.

4.4.1 Maximum Likelihood Parameter Learning

Let θ be the set of unknown CRF parameters where $\theta = \{w, v\}$. Given M i.i.d. labeled training images, the maximum likelihood estimates of the parameters are given by maximizing the log-likelihood $l(\theta) = \sum_{m=1}^{M} \log P(x^m|y^m, \theta)$, i.e.,

$$\widehat{\theta} = \underset{\theta}{\mathrm{argmax}} \sum_{m=1}^{M} \left\{ \sum_{i \in S^m} \log \sigma(x_i^m w^T h_i(y^m)) + \sum_{i \in S^m} \sum_{j \in \mathcal{N}_i} x_i^m x_j^m v^T \mu_{ij}(y^m) - \log Z^m \right\}, \tag{4.10}$$

where the partition function for the m^{th} image is,

$$Z^m = \sum_{x} \exp \left\{ \sum_{i \in S^m} \log \sigma(x_i w^T h_i(y^m)) + \sum_{i \in S^m} \sum_{j \in \mathcal{N}_i} x_i x_j v^T \mu_{ij}(y^m) \right\}.$$

Note that Z^m is a function of the parameters θ and the observed data y^m. For learning the parameters using gradient ascent, the derivatives of the log-likelihood are

$$\frac{\partial l(\theta)}{\partial w} = \frac{1}{2} \sum_{m} \sum_{i \in S^m} (x_i^m - \langle x_i \rangle_{\theta;y^m}) h_i(y^m), \tag{4.11}$$

$$\frac{\partial l(\theta)}{\partial v} = \sum_{m} \sum_{i \in S^m} \sum_{j \in \mathcal{N}_i} (x_i^m x_j^m - \langle x_i x_j \rangle_{\theta;y^m}) \mu_{ij}(y^m). \tag{4.12}$$

Here $\langle \cdot \rangle_{\theta;y^m}$ denotes expectation with $P(x|y^m, \theta)$. Ignoring $\mu_{ij}(y^m)$, gradient ascent with Equation (4.12) resembles the problem of learning in Boltzmann machines.

For arbitrary graphs with loops, the expectations in Equation (4.11) and Equation (4.12) cannot be computed exactly due to the combinatorial size of the label space. Sampling procedures such as Markov Chain Monte Carlo (MCMC) can be used to approximate the true expectations. Unfortunately, MCMC techniques have two main problems: a long 'burn-in' period (which makes them slow) and high variance in estimates. Although several techniques have been suggested to approximate the expectations, let us focus on two popular methods (see [10] for other choices and a detailed comparison).

4.4.1.1 Pseudo-Marginal Approximation (PMA)

It is easy to see that if we had true marginal distributions $P_i(x_i|y,\theta)$ at each site i, and $P_{ij}(x_i,x_j|y,\theta)$ at each pair of sites i and $j \in \mathcal{N}_i$, we could compute exact expectations as:

$$\langle x_i \rangle_{\theta;y} = \sum_{x_i} x_i P_i(x_i|y,\theta) \quad \text{and} \quad \langle x_i x_j \rangle_{\theta;y} = \sum_{x_i,x_j} x_i x_j P_{ij}(x_i,x_j|y,\theta).$$

Since computing exact marginal distributions is in general infeasible, a standard approach is to replace the actual marginals by pseudo-marginals. For instance, one can use loopy Belief Propagation (BP) to get these pseudo-marginals. It has been shown in practice that for many applications loopy BP provides good estimates of the marginals.

4.4.1.2 Saddle Point Approximation (SPA)

In Saddle Point Approximation (SPA), one makes a discrete approximation of the expectations by directly using best estimates of labels at a given setting of parameters. This is equivalent to approximating the partition function (Z) such that the summation over all the label configurations x in Z is replaced by the largest term in the sum, which occurs at the most probable label configuration. In other words, if

$$\widehat{x} = \arg\max_{x} P(x|y,\theta),$$

then according to SPA,

$$Z \approx \exp\left\{ \sum_{i\in S} \log \sigma(\widehat{x}_i w^T h_i(y)) + \sum_{i\in S} \sum_{j\in\mathcal{N}_i} \widehat{x}_i \widehat{x}_j v^T \mu_{ij}(y) \right\}.$$

This leads to a very simple approximation to the expectation, i.e., $\langle x_i \rangle_{\theta;y} \approx \widehat{x}_i$. If we further assume mean-field type decoupling, i.e., $\langle x_i x_j \rangle_{\theta;y} = \langle x_i \rangle_{\theta;y} \langle x_j \rangle_{\theta;y}$, it also follows that $\langle x_i x_j \rangle_{\theta;y} \approx \widehat{x}_i \widehat{x}_j$. Readers familiar with perceptron learning rules can readily see that with such an approximation, the updates in Equation (4.11) are very similar to perceptron updates.

However, this discrete approximation raises a critical question: Will the gradient ascent of the likelihood with such gradients converge? It has been shown empirically that while the approximate gradient ascent is not strictly convergent in general, it is weakly convergent in that it oscillates within a set of good parameters, or converges to a good parameter with isolated large deviations. In fact one can show that this weak-convergence behavior is tied to the empirical error of the model [10]. To pick a good parameter setting, one can use any of the popular heuristics used for perceptron learning with inseparable data. For instance, one can let the algorithm run up to some fixed number of iterations and pick the parameter setting that minimizes the empirical error. Even though lack of strict convergence can be seen as a drawback of SPA, the main advantage of these methods is very fast learning of parameters with performance similar to or better than pseudo-marginal methods.

4.4.2 Inference

Given a new test image y, the problem of inference is to find the optimal labels x over the image sites, where optimality is defined with respect to a given cost function. Maximum A Posteriori (MAP) solution is a widely used estimate that is optimal with respect to the zero-one cost function defined as,

$$C(x,x^*) = 1 - \delta(x - x^*), \tag{4.13}$$

where x^* is the true label configuration, and $\delta(x - x^*)$ is 1 if $x = x^*$, and 0 otherwise. The MAP solution is defined as,

$$\widehat{x} = \arg\max_{x} P(x|y,\theta).$$

For binary classifications, the MAP estimate can be computed exactly for an undirected graph using the max-flow/min-cut type of algorithms if the probability distribution meets certain conditions [5]. While using the Ising MRF model, exact MAP solution can be computed if $\beta_m \geq 0$. For the CRF model, since max-flow algorithms do not allow negative interaction between the sites, the data-dependent smoothing for each clique is set to be $v^T\mu_{ij}(y) = \max\{0, v^T\mu_{ij}(y)\}$, yielding an approximate MAP solution.

An alternative to the MAP solution is the Maximum Posterior Marginal (MPM) solution which is optimal for the sitewise zero-one cost function defined as,

$$C(x,x^*) = \sum_{i \in S}(1 - \delta(x_i - x_i^*)), \tag{4.14}$$

where x_i^* is the true label at the i^{th} site. The MPM solution at each site is defined as,

$$\widehat{x_i} = \arg\max_{x_i} P_i(x_i|y,\theta), \quad \text{where} \quad P_i(x_i|y,\theta) = \sum_{x - x_i} P(x|y,\theta),$$

and $x - x_i$ denotes all the node variables except for node i. The MPM computation requires marginalization over a large number of variables which is generally NP-hard. However, as discussed before, one can use loopy BP to obtain an estimate of the MPM solution.

4.5 Extensions

A large number of extensions of the basic binary CRFs have been proposed in the literature. In the following sections, we discuss two key extensions: Multiclass CRFs to deal with multiclass labeling problems, and Hierarchical CRFs to incorporate hierarchical context in the model.

4.5.1 Multiclass CRF

There are several applications in computer vision that require the nodes in the graph to take multiple class labels. For example, in semantic scene segmentation task, one may want to assign each pixel into one of many classes such as *sky*, *water*, *grass*, etc. In the case of image denoising applied to a 256 gray-level image, each pixel may take up to 256 labels. In the part-based paradigm of object detection, usually there are more than two characteristic parts that make the full object, and the goal is to label each generic part in the scene as a specific part of the object or background.

The extension of binary CRFs to the multiclass case is relatively straightforward. The only difference in multiclass CRF formulation is that the labels at the image sites are given by $x = \{x_i\}_{i \in S}$, where $x_i \in \{1,\ldots,C\}$ and C is the number of classes. To illustrate various terms in the model, we will take the example of parts-based object detection, in which, each image site is a part and the first $(C-1)$ labels correspond to specific object parts and the C^{th} label corresponds to the background class.

Following the arguments given in Section 4.3.1 and the form of the association potential for binary CRFs (Equation (4.7)), the association potential can be easily generalized to the multiclass case as,

$$A(x_i, y) = \sum_{k=1}^{C} \delta(x_i = k) \log P'(x_i = k|y), \tag{4.15}$$

where $\delta(x_i = k)$ is 1 if $x_i = k$ and 0 otherwise. Let $h_i(y)$ be a (possibly kernelized) feature vector at each site i. Note that, in the case of object detection, the vector $h_i(y)$ encodes the appearance based features of the i^{th} part. To model $P'(x_i = k|y)$, one can simply use the multiclass version of the logistic function described for the binary CRFs in Section 4.3.1. This leads to the softmax function in the multiclass case where,

$$P'(x_i = k|y) = \begin{cases} \frac{\exp(w_k^T h_i(y))}{1+\sum_{l=1}^{C-1} \exp(w_l^T h_i(y))} & \text{if } k < C \\ \\ \frac{1}{1+\sum_{l=1}^{C-1} \exp(w_l^T h_i(y))} & \text{if } k = C. \end{cases} \tag{4.16}$$

Here, w_k are the model parameters for $k = 1 \ldots C-1$. For a C class classification problem, one needs only $C-1$ independent hyperplanes, which may lie in a high dimensional (kernel-projected) space inducing a non-linear decision boundary in the original feature space. In the application of object detection, the association potential discriminatively models the individual appearance of each part in the image.

The interaction potential in CRFs predicts how the labels at two sites interact given the observations. Generalizing the interaction potential given for binary CRFs, interaction potential for multiclass CRFs can be written as,

$$I(x_i,x_j,y) = \sum_{k=1}^{C}\sum_{l=1}^{C} v_{kl}^T \mu_{ij}(y)\delta(x_i = k)\delta(x_j = l). \tag{4.17}$$

Here, $\mu_{ij}(y)$ is the pairwise relational vector for a site pair (i,j), and v_{kl} are the model parameters. Note that in the case of object detection, the vector $\mu_{ij}(y)$ encodes the pairwise features required for forcing geometric and possibly photometric consistency in the pair of parts. For undirected graphs, the site pairs are unordered sets implying that $v_{kl} = v_{lk}$ for $k,l = 1\ldots C$. The from of interaction potential given in Equation (4.17) is a generalization of the Potts model used commonly in computer vision problems such as image segmentation and restoration. The standard Potts model can be recovered from Equation (4.17) if $v_{kl} = 0$ when $k \neq l$, and all the elements of the vector v_{kl} are set to zero except the bias term. A more specific but popular form of Potts model is achieved if the bias terms for all the vectors v_{kk} $\forall$ k are also fixed to be the same. Similar to the interaction potential of the binary CRF, multiclass interaction potential can be seen as a pairwise discriminative model which partitions the pairwise relational feature space (induced by the features $\mu_{ij}(y)$) in $C(C+1)/2$ regions.

It is important to note that, to enforce the geometric consistency relationship between parts, the interaction between part labels has to use observed data (e.g. the location of patches). Since, the pairwise potential I is a function of observed data in CRFs, these fields provide a principled way to represent relations between parts in a random-field framework.

Let θ be the set of CRF parameters where $\theta = \big\{\{w_k\}_{k=1,\ldots,C-1}, \{v_{kl}\}_{k,l=1,\ldots,C}\big\}$. To learn θ via maximum likelihood, similar to the binary CRFs, one can write the gradient of log-likelihood as,

$$\frac{\partial l(\theta)}{\partial w_k} = \sum_m \sum_{i\in S^m} \Big(\delta(x_i^m = k) - \langle\delta(x_i = k)\rangle\Big) h_i(y^m), \tag{4.18}$$

$$\frac{\partial l(\theta)}{\partial v_{kl}} = \sum_m \sum_{i\in S^m} \sum_{j\in\mathcal{N}_i} \Big(\delta(x_i^m = k)\delta(x_j^m = l) - \big\langle\delta(x_i = k)\delta(x_j = l)\big\rangle\Big)\mu_{ij}(y^m), \tag{4.19}$$

where $\langle.\rangle$ denotes expectation with respect to the distribution $P(x|y^m,\theta)$ and m indexes over the training images. Generally the expectations in Equation (4.18) and Equation (4.19) cannot be computed exactly even for moderate-size problems. Similar to the previous discussion in Section 4.4, these expectations can be approximated by either pseudo-marginals or Saddle Point Approximation with multiclass extensions of min-cuts [3]. Similarly, for inference, one can get the labels either using approximate MAP obtained by multiclass min-cut or using approximate MPM via loopy BP.

4.5.2 Hierarchical CRF

So far, we have discussed spatial interactions in natural images at pixel, block or patch level for binary or multiclass classification problems. However, in natural

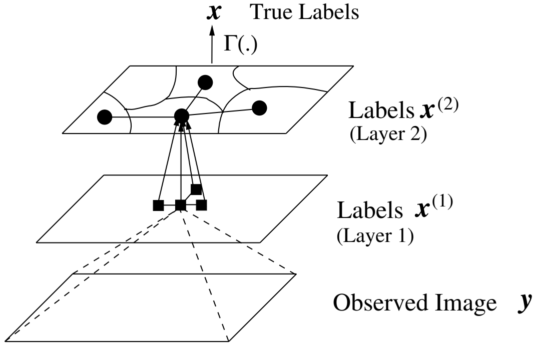


Fig. 4.6 A simple illustration of a two-layer hierarchical field for contextual classification. Squares and circles represent sites at the two layers. Only one node along with its neighbors is shown for each layer for clarity. Layer 1 models short-range interactions while layer 2 models long range dependencies in images. The true labels x are obtained from the top layer by a simple replication mapping $\Gamma(.)$.

images, there are different levels of context one would like to use to improve classification accuracy. For instance, for pixelwise image labeling problem, the local smoothness of pixel labels provides local context. On the other hand, there exists a higher level global context since image regions follow probable configurations. For example, sky tends to occur above water or vegetation. Similarly, for the problem of parts-based object detection, local context is the geometric relationship among parts of an object while the relative spatial configurations of different objects (e.g., monitor, keyboard and mouse) provides the global context. Here we present a high-level discussion on how one can use hierarchy of CRFs to improve classification in images. For a detailed discussion on this topic, see [13].

A simple two-layer hierarchical model is shown in Figure 4.6, in which each layer is modeled as a separate CRF. The first layer models short range interactions among the sites such as label smoothing for pixelwise labeling, or geometric consistency among parts of an object. The second layer models the long range interactions between groups of sites corresponding to different coherent regions or objects. Thus, this layer can take into account interactions between different objects (monitor/keyboard) or regions (sky/water).

The two layers of the hierarchy are coupled with directed links. A node in layer 1 may represent a single pixel or a patch while a node in layer 2 represents a larger homogeneous region or a whole object. Each node in the two layers is connected to its neighbors through undirected links. In addition, each node in layer 2 is also connected to multiple nodes in layer 1 through directed links. The use of directed links between the two layers, instead of the undirected ones, avoids the intractability of dealing with a large partition function. Each layer being a CRF, any node in layer 1 can potentially use arbitrary features from the whole image. The top layer uses the output of layer 1 as input through the directed links.

Given the observed data $y = \{y_i\}_{i \in S}$ in an image, we are interested in finding the labels, $x = \{x_i\}_{i \in S}$, where $x_i \in \mathcal{L}$ and $|\mathcal{L}|$ is the number of classes. For image labeling, a site is a pixel and a class may be *sky*, *grass*, etc., while for contextual object detection, a site is a patch and a class may refer to objects (e.g., *keyboard* or *mouse*). The set of sites in layer 1 is $S^{(1)}$ such that $S^{(1)} = S$, while that in layer 2 is denoted by $S^{(2)}$. The nodes in layer 2 induce a partition over the set $S^{(1)}$ such that a subset of nodes in layer 1 correspond to one node in layer 2. Formally, a partition h is defined as $h : S^{(1)} \rightarrow S^{(2)}$ such that, if $S_r^{(1)}$ is a subset of nodes in layer 1 corresponding to node $r \in S^{(2)}$, then $S^{(1)} = \bigcup\limits_r S_r^{(1)}$ and $S_r^{(1)} \cap S_s^{(1)} = \phi \quad \forall\ r,s \in S^{(2)}$. Let the space of all partitions be denoted as $\mathcal{H}$. This partition should not be confused with an image partition, since it is defined over the sites in $S^{(1)}$, which may not correspond to the image pixels (e.g., in object detection, where sites are random image patches). Let the labels on the sites in the two layers be given by $x^{(1)} = \{x_i^{(1)}\}_{i \in S^{(1)}}$ and $x^{(2)} = \{x_r^{(2)}\}_{r \in S^{(2)}}$, where $x_i^{(1)} \in \mathcal{L}^{(1)}$ and $x_r^{(2)} \in \mathcal{L}^{(2)}$, where $\mathcal{L}^{(2)} = \mathcal{L}$. The nodes in layer 1 may take pseudo-labels that are different from the final desired labels. For instance, in object detection, a node at layer 1 may be labeled as 'a certain part' of an object rather than the object itself. In fact, the labels at this layer can be seen as noisy versions of the true desired labels.

Given an image y, we are interested in obtaining the discriminative distribution $P(x|y)$ over the true labels. Let us define a space of valid partitions, $\mathcal{H}_v$, such that $\forall\ \ h \in \mathcal{H}_v,\ x_i = x_r^{(2)} \quad \forall\ i \in S_r^{(1)}$, where $r = h(i)$. This implies that multiple nodes in layer 1 form a hypothesis about a single *homogeneous* region or an object in layer 2. Further, we define a replication mapping, $\Gamma(.)$, which takes any value (discrete or continuous) on node r and assigns it to all the nodes in $S_r^{(1)}$. Thus, given a partition $h \in \mathcal{H}_v$, and the corresponding labels $x^{(2)}$, the labels x can be obtained simply by replication. This implies, $P(x|y) \equiv P(x^{(2)}|h,y)$ if $h \in \mathcal{H}_v$. However, given an observed image y, the constraint $h \in \mathcal{H}_v$ is too restrictive. Instead, one can define a distribution, $P(h|y)$, that prefers partitions in $\mathcal{H}_v$ over all possible partitions, and,

$$\begin{aligned} P(x|y) &\cong \sum_{h \in \mathcal{H}} P(x^{(2)}|h,y)P(h|y) \\ &= \sum_{h \in \mathcal{H}} \sum_{x^{(1)}} P(x^{(2)}|h,x^{(1)})P(h|x^{(1)})P(x^{(1)}|y), \end{aligned} \tag{4.20}$$

where both $P(x^{(1)}|y)$ and $P(x^{(2)}|h,x^{(1)})$ are modeled as CRFs. In Equation (4.20), computing the sum over all the possible configurations of $x^{(1)}$ is a NP-hard problem. One naive way to reduce the complexity is to do inference in layer 1 until equilibrium is reached and then use this configuration $\widehat{x}^{(1)}$ as input to the next layer, i.e., $P(x^{(1)}|y) = \delta(x^{(1)} - \widehat{x}^{(1)})$. However, by doing this, one loses the power of modeling the uncertainty associated with the labels in layer 1, which was included explicitly in Equation (4.20) through $P(x^{(1)}|y)$. Here, we discuss a simple variant, where along with the equilibrium configuration, one also propagates the uncertainty associated with it to the next layer. The sitewise maximum marginal configuration are used as $\widehat{x}^{(1)}$. Let the marginals at each site i be $b_i(x_i^{(1)}) = \sum_{x^{(1)} - x_i^{(1)}} P(x^{(1)}|y)$, and $b(x^{(1)}) = \{b_i(x_i^{(1)})\}_{i \in S^{(1)}}$. The belief set, $b(x^{(1)})$ is propagated as an input to the next layer. Since the configuration $\widehat{x}^{(1)}$ can be obtained directly from $b(x^{(1)})$ by taking its sitewise maximum configuration, we omit explicit conditioning on $\widehat{x}^{(1)}$. Thus,

$$P(x|y) \approx \sum_{h\in\mathcal{H}} P(x^{(2)}|h,b(x^{(1)}))P(h|b(x^{(1)})). \tag{4.21}$$

Note that both terms in the summation implicitly include the transition probabilities $P(x_r^{(2)}|\widehat{x}_i^{(1)})$. For the first term, these are absorbed in the unary potential of the CRF in layer 2.

The model describing layer 1 is a simple multiclass CRF and different potentials are designed as discussed before in Section 4.5.1. The CRF formulation for layer 2 can be obtained in the same way except that the observations for this layer are $b(x^{(1)})$, the set of sites is $S^{(2)}$, and the label set is $\mathcal{L}^{(2)}$. The only difference lies in the form of the unary potential. Each node $r \in S^{(2)}$ in this layer receives beliefs as input from the nodes contained in set $S_r^{(1)}$ from the layer below. Taking into consideration the transition probabilities on the directed links between node r and all the nodes in $S_r^{(1)}$, the unary potential can be written as,

$$A^{(2)}(x_r^{(2)},b(x^{(1)})) = \sum_{k\in\mathcal{L}^{(2)}} \Big\{\delta(x_r^{(2)} = k)$$

$$\Big(\log P'(x_r^{(2)} = k|b(x^{(1)})) + \frac{1}{|S_r^{(1)}|}\sum_{i\in S_r^{(1)}} \log P(x_r^{(2)} = k|\widehat{x}_i^{(1)})\Big)\Big\}. \tag{4.22}$$

Here, the first term in parentheses on the right hand side involves local classifier $P'(.)$, which is again modeled as a softmax function. It takes features as input, which are constructed using the beliefs $b(x^{(1)})$ at layer 1. The second term arises due to the directed connections between each node $r \in S^{(2)}$ in layer 2 to all the nodes in the set $S_r^{(1)}$ in layer 1. The effect of this term can be understood by switching the first term off along with the interaction potential. This will lead to the intuitive reasoning of assigning node r that label which maximizes the joint transition probability (computed by assuming each site in $S_r^{(1)}$ to be independent) given a label configuration $\widehat{x}^{(1)}$ at layer 1. The term, $|S_r^{(1)}|$ acts as a normalizer that takes into account the different cardinalities of sets $S_r^{(1)}$. In the interaction potential for this layer, the features $\mu_{ij}(.)$ are designed such that they capture relative configurations of two regions or objects.

The distribution $P(h|b(x^{(1)}))$ indicates *goodness* of a partition in layer 2. Here, we just mention that one can design this function according to the application domain. One can find more details on possible choices in [13]. The set of parameters Θ, to be learned in the hierarchical model, includes the parameters of the CRFs at layer 1 and layer 2, and the transition probability matrices $P(x_r^{(2)}|\widehat{x}_i^{(1)})$. The CRF parameters for each layer are the parameters of the unary and pairwise potentials i.e., $\theta^{(\alpha)} = \left\{w_k^{(\alpha)}, v_{kl}^{(\alpha)}\right\}_{\forall k,l}^{\alpha=1,2}$. The parameters in the joint model are learned sequentially using loopy BP. The procedure is a simple extension of learning in multiclass

CRFs discussed before. Similarly, inference in this model is carried out using a combination of loopy BP and sampling of partitions. We refer the reader to [13] for more details on learning and inference.

4.6 Applications

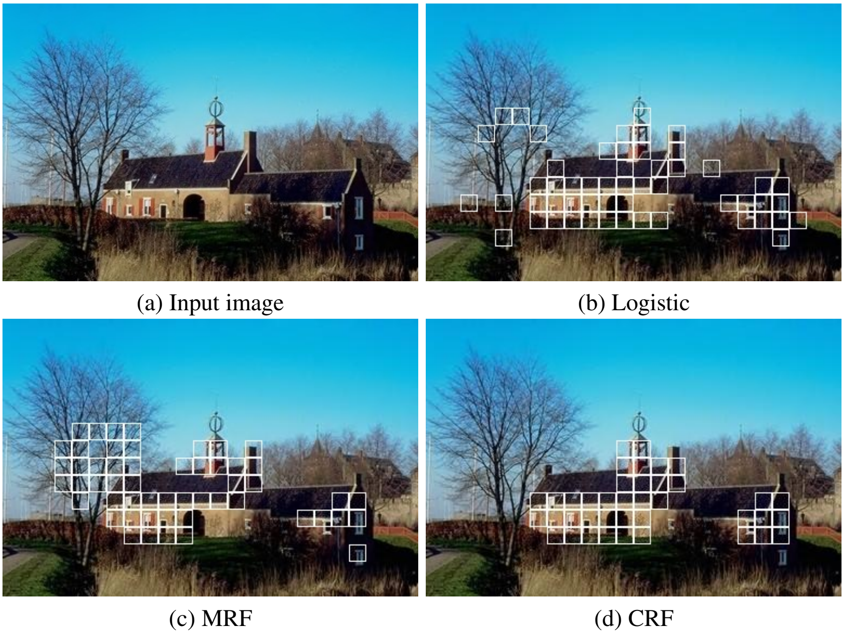
In this Section, we discuss a few real-world applications of different types of CRFs. The application of binary CRFs is considered on man-made structure detection, while the performance of multiclass and hierarchical CRFs is tested on image classification and contextual object detection tasks.

4.6.1 Man-Made Structure Detection

Detecting man-made structures in natural scenes is difficult because there are significant within class variations in the appearance of data from the *structured* class. Similarly, the data from the *nonstructured* class is virtually unconstrained, and there is a large overlap between these two classes. The training and the test set used in this study contained 108 and 129 images respectively from the Corel image database. Each image is divided into nonoverlapping 16×16 pixels blocks. Each block forms a site in the graph. The whole training set contained $36,269$ blocks from the *nonstructured* class, and $3,004$ blocks from the *structured* class.

To generate the features, the intensity gradients contained within a window in the image are combined to yield a histogram over gradient orientations. Each histogram count is weighted by the gradient magnitude at that pixel and smoothed using kernel smoothing. Heaved central-shift moments are computed to capture the the average *spikeness* of the smoothed histogram as an indicator of the *structuredness* of the patch. The orientation based feature is obtained by passing the absolute difference between the locations of the two highest peaks of the histogram through sinusoidal nonlinearity. The absolute location of the highest peak is also used.

For each image, two different type of feature vectors at each site are computed. First a *single-site* feature vector at the site i, $s_i(y_i)$ is computed using the histogram from the data y_i at that site. Obviously, this vector does not take into account the influence of the data in the neighborhood of that site. The vector $s_i(y_i)$ is composed of the first three moments and the two orientation based features described above. Next, a *multiscale* feature vector at the site i, $f_i(y)$ is computed which explicitly takes into account the dependencies in the data contained in the neighboring sites. It should be noted that the neighborhood for the data interaction need not be the same as for the label interaction. To compute $f_i(y)$, smoothed histograms are obtained at three different scales, where each scale is defined as a varying window size around the site i. The number of scales is chosen to be 3, with the scales changing in regular octaves. The lowest scale is fixed at 16×16 pixels (i.e., the size of a single site), and the highest scale at 64×64 pixels. The moment and orientation based features are obtained at each scale similar to $s_i(y_i)$. In addition, two interscale features are also obtained using the highest peaks from the histograms at consecutive scales. To avoid



(a) Input image

(b) Logistic

(c) MRF

(d) CRF

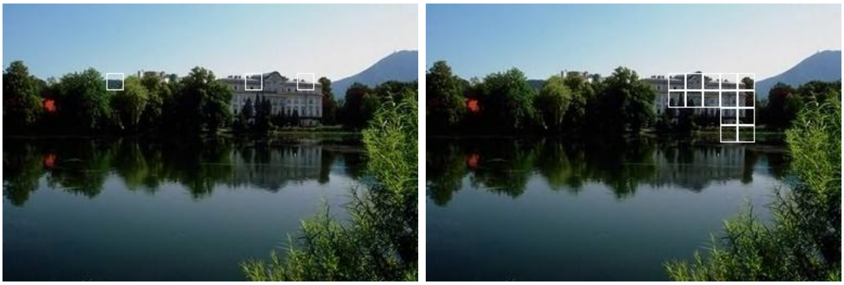
Fig. 4.7 Structure detection results on a test example for different methods. For similar detection rates, CRF reduces the false positives considerably.

redundancy in the moments based features, only two moment features are used from each scale yielding a 14 dimensional feature vector.

To make the unary classifier in the CRF more powerful, a transformed feature vector $h_i(y)$ is computed at each site i by using an explicit quadratic kernel. The quadratic mapping gives a 119 dimensional vector at each site. The features ψ_i are chosen to be the same as f_i. Further, the pairwise data vector $\mu_{ij}(y)$ is obtained by concatenating $\psi_i(y)$ and $\psi_j(y)$. The parameters of the CRF model were learned using the maximum likelihood framework as described before.

4.6.1.1 Qualitative Evaluation

For an input test image given in Figure 4.7 (a), the *structure* detection results from three methods are shown in Figure 4.7. The blocks identified as *structured* have been shown enclosed within an artificial boundary. It can be noted that for similar detection rates, the number of false positives have significantly reduced for the CRF based detection. Locally, different branches may yield features similar to those from the man-made structures. The logistic classifier does not enforce smoothness in labels, which led to increased isolated false positives. However, the MRF solution with Ising model simply smooths the labels without taking observations into account resulting in a smoothed false positive region around the tree branches.



(a) MRF

(b) CRF

Fig. 4.8 Detection of a building in poor illumination conditions in a test image. CRFs can improve the detection rate while simultaneously reducing the false positive rate.

The performance of MRF and CRF is compared on another test example requiring detection of a building in poor illumination conditions (Figure 4.8). CRFs give higher detection rate while reducing the false positive rate by enforcing interactions among the labels as well as the data from multiple scales.

4.6.1.2 Quantitative Evaluation

To carry out the quantitative evaluations, the detection rates and the number of false positives per image for each technique are compared. To avoid the confusion due to different effects in the CRF model, the first set of experiments is conducted using the *single-site* features for all the three methods. Thus, no neighborhood data interaction is used for both the logistic classifier and the CRF, i.e., $f_i(y) = s_i(y)$. The comparative results for the three methods are given in Table 4.1 next to 'MRF', 'Logistic$^-$' and 'CRF$^-$'. For comparison purposes, the false positive rate of the logistic classifier is fixed to be the same as the CRF in all the experiments. It can be noted that for similar false positives, the detection rates of the traditional MRF and the CRF are higher than the logistic classifier due to the label interaction. However, the higher detection rate of the CRF in comparison to the MRF indicates the gain due to the use of discriminative models in the association and interaction potentials. In the next experiment, to take advantage of the power of the CRF framework, data interaction was allowed for both the logistic classifier as well as the CRF ('Logistic' and 'CRF' in Table 4.1). The CRF detection rate increases substantially and the false positives decrease further indicating the importance of allowing the data interaction in addition to the label interaction.

4.6.2 Image Classification and Contextual Object Detection

The experiments in this Section demonstrate the capability of hierarchical CRFs. The first set of experiments is based on the 'Beach' dataset from [14], which contains a collection of consumer photographs. The goal was to assign to each image pixel one of 6 class labels: $\{$*sky*, *water*, *sand*, *skin*, *grass*, *other*$\}$. This dataset is

Table 4.1 Detection Rates (DR) and False Positives (FP) for the test set containing 129 images (49,536 sites). FP for logistic classifier were kept to be the same as for CRF for DR comparison. Superscript $'-'$ indicates no neighborhood data interaction was used.

	MRF	Logistic$^-$	CRF$^-$	Logistic	CRF
DR (%)	58.35	47.50	61.79	60.80	72.54
FP (per image)	2.44	2.28	2.28	1.76	1.76

particularly challenging due to wide within-class variance in the appearance of the data due to drastic illumination conditions. Another characteristic of this dataset which makes it difficult is that, for most of the images, a significant area belongs to none of the semantic classes (i.e., falls under the *other* category). Traditionally it has been hard to model this category because any pixel in this category can virtually have unconstrained appearance.

The layer 1 of hierarchical CRF implements smoothness of pixel labels as the local context. Hence, the sites in layer 1 are image pixels, and three HSV color features and two texture features (based on second moment matrix) give a 5 dimensional unary feature vector. Use of quadratic kernel yielded a 21 dimensional feature vector $h_i(y)$. To implement label smoothing, the pairwise feature vector $\mu_{ij}(y)$ is set to 1, resulting in the Potts model.

The layer 2 encodes interactions among different regions given the beliefs at layer 1 and a partition. Each region of the partition is a site in layer 2. For this dataset, the number of sites at layer 2 varied from 13 to 49 for different images. Each node in this layer is connected to every other node. The computations over these complete graphs are still efficient because of the small number of nodes in the graph. The unary feature vector for each node r consists of normalized product of beliefs from all the sites i in $S_r^{(1)}$ and the normalized centroid location of the region r. This gives an 8 dimensional feature vector. Further, quadratic transforms are used to obtain a 44 dim vector. The pairwise features between regions are chosen to be binary indicator attributes: a region is *above*, *beside* or *enclosed* within another region.

A few example results from the test set are shown in Figure 4.9. The softmax classifier (second column) does not perform well because it classifies each pixel independently without considering interactions in the labels. For example, there is substantial confusion between sand and skin regions or water and sky regions. In addition, the labels are not smooth giving the resulting classification a dithered appearance. The smoothness of labels can be achieved (third column) by implementing smoothing interaction potential in layer 1 of the hierarchical CRF. However, the errors in the larger regions are not eliminated. But, when the full hierarchical model is applied where the second layer enforces the spatial configuration of the regions, most of the errors are eliminated. Note that there are several images that contain pixels which do not belong to any of the semantic classes (e.g., clothing, chairs, boat etc). Such regions are also classified well by the hierarchical CRF. The last row in Figure4.9 shows that the average accuracy on the test set increases to 74% using the full hierarchical model in comparison to 62.3% from the softmax classifier.

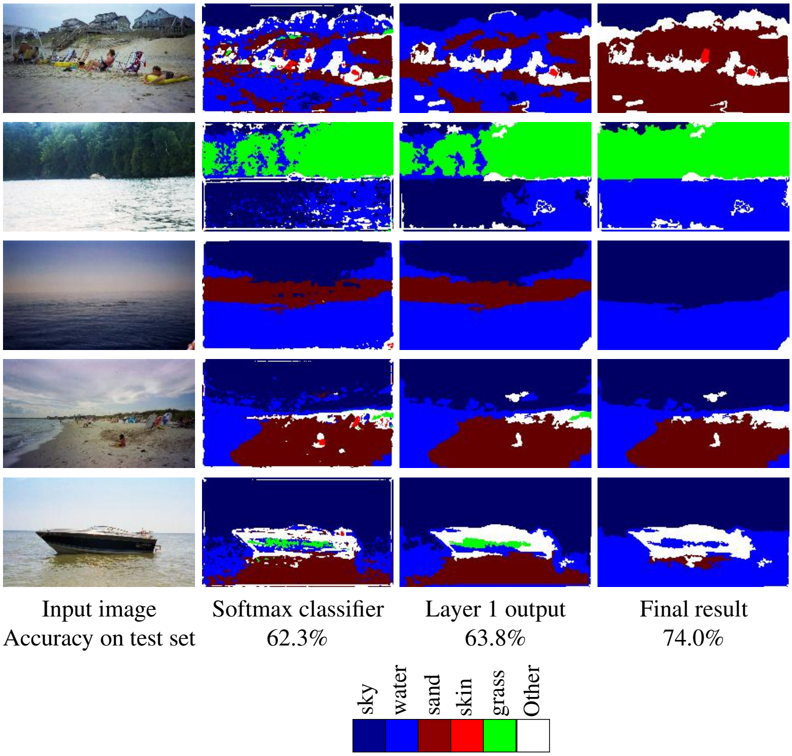


Fig. 4.9 Pixelwise classification results on the Beach dataset using hierarchical CRFs. 'Layer 1 output' shows the result of implementing label interactions through layer 1 only. Label smoothing is achieved but many large regions are labeled wrong in this output. 'Final result' shows the final classification using both the layers in the hierarchical model which eliminates most of the errors.

The second set of experiments aims to detect objects i.e., monitor, keyboard and mouse in an office scene. The dataset contained 164 low-resolution images of size less than 100 × 100 pixels each [21]. The main challenge in the dataset is the detection of the keyboard and the mouse, which spanned only a few pixels in the images. For these experiments, the hierarchical CRF enforces interactions among the three objects, resulting in a significant reduction in false alarms.

For each object, at first a base detector is trained using gentle-boost. Since the size of the mouse in the input images is very small (on average about 8 × 5 pixels), the boosting based detector could not be trained for the mouse. Instead, a simple template matching based detector is learned. A patch at a location, where the output of any of the three detectors is higher than a threshold, represents a site in $S^{(1)}$. The set of sites $S^{(2)}$ in layer 2 is the same as in layer 1, indicating the trivial partition. The label set for the sites in $S^{(1)}$ and $S^{(2)}$ is $\{monitor, keyboard, mouse, background\}$. Since layer 1 uses the output of a standard object detector, interactions among sites take place only at layer 2.

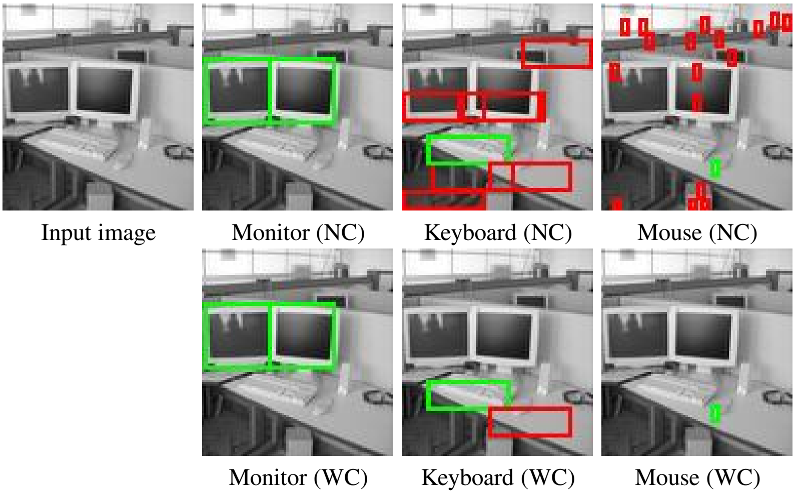

Input image
Monitor (NC)
Keyboard (NC)
Mouse (NC)
Monitor (WC)
Keyboard (WC)
Mouse (WC)

Fig. 4.10 Detection results for monitor, keyboard and mouse using context based on spatial configuration of objects. NC - No Context, WC - With Context. Monitor detection was good with the base detector itself due to less appearance ambiguity. Note the impoverished appearances of the keyboard and the mouse. Green and red indicate true detections and false alarms respectively.

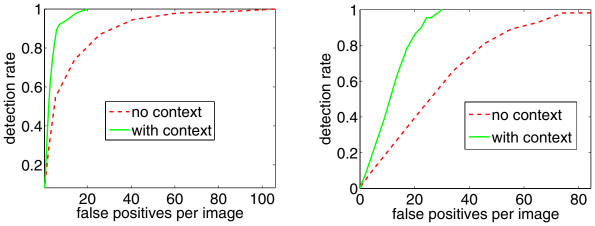


Fig. 4.11 The ROC curves for the detection of keyboard (left) and mouse (right). Relatively high false alarm rates for the mouse were due to very small size of mouse (about 8×5 pixels) in the input images.

The unary features at layer 2 consist of the score from each detector yielding a 3 dimensional feature vector. The difference of coordinates of the patch centers resulted in a 2 dimensional pairwise feature vector. Each node is connected to every other node in this layer. Figure 4.10 shows a typical result from the test set. It is clear that the false alarms are reduced considerably in comparison to the base detector. The use of context did not change the results for the monitor, since the base detector itself was able to give good performance. This is reasonable because one hopes the context to be more useful when the local appearance of an object is more ambiguous. The ROC curves for the keyboard and the mouse detection are compared with the corresponding base detectors in Figure 4.11. For the mouse detection, even though the hierarchical CRF was able to reduce the false positives significantly, the number of false alarms per image is still high. This is understandable because the size of mouse is very small in all the images. One can hope for context to improve detection only if there exists at least 'bare-minimum' appearance based evidence for that object in images.

4.7 Related Work and Further Readings

In this chapter, we gave a succinct review of basic types of CRFs used in computer vision. One can find a more in-depth discussion on modeling, parameter learning and inference in these CRFs in [9]. It also contains extensive details on the experimental procedures including feature extraction and speed comparisons.

CRFs were introduced in computer vision by Kumar and Hebert [11] [12] extending the 1D-CRFs from Lafferty et al. [15]. Since then, a number of techniques have been proposed in vision that further modified CRFs for various applications. Different types of local classifiers such as neural network [7], boosted stumps [21] and probit function [19] have been used to model clique potentials. A Hidden CRF model was introduced in [20] to handle latent variables. Learning in CRFs was extended to a semi-supervised paradigm by [17]. As a final note, we would like to mention that taking a non-probabilistic view, energy based models have been used in vision. These models have expressive power similar to CRFs [2] [16]. However, effective parameter learning is perhaps the biggest challenge in such non-probabilistic models.

References

1. Besag, J.: On the statistical analysis of dirty pictures. Journal of Royal Statistical Society B-48, 259–302 (1986)
2. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In: Proceedings of the International Conference on Computer Vision (ICCV), vol. I, pp. 105–112 (2001)
3. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE transactions on Pattern Analysis and Machine Intelligence 23(11), 1222–1239 (2001)
4. Geman, S., Geman, D.: Stochastic relaxation, gibbs distribution and the bayesian restoration of images. IEEE transactions on Pattern Analysis and Machine Intelligence 6, 721–741 (1984)
5. Greig, D.M., Porteous, B.T., Seheult, A.H.: Exact maximum a posteriori estimation for binary images. Journal of Royal Statistical Society 51(2), 271–279 (1989)
6. Hammersley, J.M., Clifford, P.: Markov field on finite graph and lattices (unpublished)
7. He, X., Zemel, R., Carreira-Perpinan, M.: Multiscale Conditional Random Fields for image labelling. In: Proceedings of International Conference on Computer Vision and Pattern Recognition (2004)
8. Ising, E.: Beitrag zur theorie der ferromagnetismus. Zeitschrift Fur. Physik. 31, 253–258 (1925)

9. Kumar, S.: Models for Learning Spatial Interactions in Natural Images for Context-Based Classification. PhD Thesis, Carnegie Mellon University, The Robotics Institute, School of Computer Science (2005)
10. Kumar, S., August, J., Hebert, M.: Exploiting inference for approximate parameter learning in discriminative fields: An empirical study. In: Rangarajan, A., Vemuri, B.C., Yuille, A.L. (eds.) EMMCVPR 2005. LNCS, vol. 3757, pp. 153–168. Springer, Heidelberg (2005)
11. Kumar, S., Hebert, M.: Discriminative fields for modeling spatial dependencies in natural images. In: Proceedings of the International Conference on Neural Information Processing Systems, NIPS (2003)
12. Kumar, S., Hebert, M.: Discriminative Random Fields: A discriminative framework for contextual interaction in classification. In: Proceedings of the International Conference on Computer Vision (ICCV), vol. 2, pp. 1150–1157 (2003)
13. Kumar, S., Hebert, M.: A hierarchical field framework for unified context-based classification. In: Proceedings of the International Conference on Computer Vision, ICCV (2005)
14. Kumar, S., loui, A.C., Hebert, M.: An observation-constrained generative approach for probabilistic classification of image regions. Image and Vision Computing, Special Issue on Generative Models Based Vision 21, 87–97 (2003)
15. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the International Conference on Machine Learning (2001)
16. LeCun, Y., Huang, F.J.: Loss functions for discriminative training of energy-based models. AI-Stats (2005)
17. Lee, C., Wang, S., Jiao, F., Schuurmans, D., Greiner, R.: Learning to model spatial dependency: semi-supervised discriminative random fields. In: Proceedings of the International Conference on Neural Information Processing Systems Conference, NIPS (2006)
18. McCullagh, P., Nelder, J.A.: Generalised Linear Models. Chapman and Hall, London (1987)
19. Qi, Y., Szummer, M., Minka, T.P.: Diagram structure recognition by Bayesian Conditional Random Fields. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition, CVPR (2005)
20. Quattoni, A., Collins, M., Darrell, T.: Conditional Random Fields for object recognition. In: Proceedings of the International Conference on Neural Information Processing Systems, NIPS (2004)
21. Torralba, A., Murphy, K.P., Freeman, W.T.: Contextual models for object detection using Boosted Random Fields. In: Proceedings of the International Conference on Neural Information Processing Systems, NIPS (2005)