Conditional Random Fields
1. Generative Model vs Discriminative Model

Chunpai Wang

June 06, 2017

This note mainly refers to the monograph "An Introduction to Conditional Random Fields" by Sutton and McCallum [1].

1 Introduction

Structured prediction methods are essentially a combination of classification and graphical modeling. They combine the ability of graphical models to compactly model multivariate data with the ability of classification methods to perform prediction using large sets of input features. Conditional random fields is a popular probabilistic method for structured prediction. This note will cover methods for inference and parameter estimation for CRFs, including practical issues for implementing large-scale CRFs.

In many applications, we wish to predict an output vector $\mathbf{y} = \{y_0, y_1, ..., y_T\}$ of random variables given an observed feature vector $\mathbf{x}$. The difference between generative models and CRFs is thus exactly analogous to the difference between the naive Bayes and logistic regression classifiers. Indeed, the multinomial logistic regression model can be seen as the simplest kind of CRF, in which where is only one output variable.

2 Graphical Modeling

It is difficult and expensive to represent distributions over many variables. The insight of the graphical modeling perspective is that a distribution over very many variables can often be represented as a product of local functions that each depend on a much smaller subset of variables. Graphical model is used to encode relationships between variables, and graph represents a set of independences and factorizes distribution. This **factorization** turns out to have close connection to certain conditional independence relationships among the variables - both types of information being easily summarized by a graph. Conditional independence will make the representation of conditional probabilities "shorter", which means it has less number of variables interacted. We will learn the graphical models from both the factorization and conditional independence viewpoints.

2.1 Undirected Model

Consider a set of random variables Y. The undirected graph $G(V, E)$ represents the probability distribution of $p(Y)$. In the graph G, each node represents a random variable, and each edge represents the dependency between two random variables. If the joint distribution $p(Y)$ satisfy the pairwise Markov property, local Markov property, or global Markov property, then we call this joint distribution as Markov random field.

Every variable $Y_s \in Y$ where $s \in \{1, 2, ..., |Y|\}$ takes outcomes from a set $\mathcal{Y}$, which can be either continuous and discrete. An arbitrary assignment to Y is denoted by a vector $\mathbf{y}$. Given a variable $Y_s \in Y$, the notation y_s denotes the value assigned to Y_s by $\mathbf{y}$. For example, we have a picture with $10 * 10$ pixels, so we can view this picture as 100 random variables, and each pixel is a random variable which can take different values.

The notation $\mathbb{1}_{\{y=y'\}}$ denotes an indicator function of y which takes the value 1 when $y = y'$ and 0 otherwise. We also require notation for marginalization. For a fixed variable assignment y_s, we use the summation $\sum_{\mathbf{y} \backslash y_s}$ to indicate a summation over all possible assignments $\mathbf{y}$ whose value for variable Y_s is equal to y_s.

Suppose a probability distribution p of interest can be represented by a product of A number of factors of the form $\Psi_a(\mathbf{y}_a)$, where $a \in \{1, 2, ..., A\}$ and each factor $\Psi_a(\mathbf{y}_a)$ depends only on a subset of $Y_a \subseteq Y$ of the variables. One advantage is Y_a may be much smaller than the full variable set Y, which allows us to reprenet p much more efficiently. The value $\Psi_a(\mathbf{y}_a)$ is a non-negative scalar that can be thought of as a measure of how compatible the value $\mathbf{y}_a$ are with each other. Assignments that have higher compatibility values will have higher probability.

Formally, given a collection of subsets $\{Y_a\}_{a=1}^A$ of Y, an undirected graphical model is the set of all distributions that can be written as

$$p(\mathbf{y}) = \frac{1}{Z}\prod_{a=1}^{A}\Psi_a(\mathbf{y}_a) \tag{1}$$

for any choice of factors $F = \{\Psi_a\}_{a=1}^A$ that have $\Psi_a(\mathbf{y}_a) \geq 0$ for all $\mathbf{y}_a$. The factors are also called local functions or compatibility functions. The constant Z is a normalization factor that ensures the distribution p sums to 1. It is defined as

$$Z = \sum_{\mathbf{y}}\prod_{a=1}^{A}\Psi_a(\mathbf{y}_a) \tag{2}$$

where the summation is over the exponentially many possible assignments to $\mathbf{y}$. For this reason, computing Z is intractable in general, but much work exists on how to approximate it.

$\mathbf{y}$ is the an assignment of random variables Y. Thus, undirected graphical model can easily define the joint probability of an assignment of a set of random variables, with some assumptions that some near neighbors are dependent with each other. We care about how to compute the joint distribution. For a given probabilistic undirected model, we would like to factor the joint distribution, which will benefit the learning and inference of the model. In fact, one main advantage of undirected graph is it is easy to get factor graph, which can help write the joint distribution into factor form.

2.2 Factor Graph

The reason for the term "graphical model" is that the factorization (1) can be represented compactly by means of a graph, such as factor graphs. A factor graph is a bipartite graph $G = (V, F, E)$ in which one set of $V = \{1, 2, ..., |Y|\}$ indexes the random variables in the model, and the other set of nodes $F = \{1, 2, ..., A\}$ indexes the factors. If a variable node Y_s for $s \in V$ is connected to a factor node Ψ_a for $a \in F$, then Y_s is one of the argument of Ψ_a. Now, we need to check if a factor graph "describes" a given distribution or not.

Definition 1. A distribution $p(\mathbf{y})$ factorizes according to a factor graph G if there exists a set of local function Ψ_a such that p can be written as

$$p(\mathbf{y}) = \frac{1}{Z}\prod_{a\in F}\Psi_a(\mathbf{y}_{N(a)}) \tag{3}$$

where $N(a)$ denotes the neighbors of the factor with index a, i.e., a set of variable indices.

Remark: in other words, if a distribution $p(\mathbf{y})$ can be represented in the form of (1),then it can be described by a factor G.

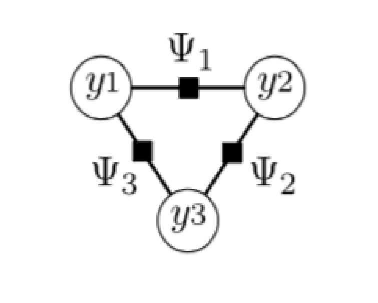

Figure 1: An example of a factor graph over three variables. This factor graph describes the set of all distributions p over 3 variables that can be written as $p(y_1, y_2, y_3) = \Psi_1(y_1, y_2)\Psi_2(y_2, y_3)\Psi_3(y_1, y_3)$ for all $\mathbf{y} = (y_1, y_2, y_3)$

2.3 Markov Network

Markov Network are graphs over random variables only, rather than random variables and factors. It directly represents conditional independence relationships in a multivariate distribution.

Let G be an undirected graph over integers $V = \{1, 2, ..., |Y|\}$ that index each random variable of interest. For a variable index $s \in V$, let $N(s)$ denote it neighbors in G. Then we say that a distribution p is Markov with respect to G if it satisfies the local Markov property: for any two variables $Y_s, Y_t \in Y$m the variable Y_s is independent of Y_t conditioned on its neighbors $Y_{N(s)}$. Intuitively, this means that $Y_{N(s)}$ on its own contains all of information that is useful for predicting Y_s. We can see this is very similar to the idea of applying convolution on small region of image.

A Markov network has an undesirable ambiguity from the factorization perspective.

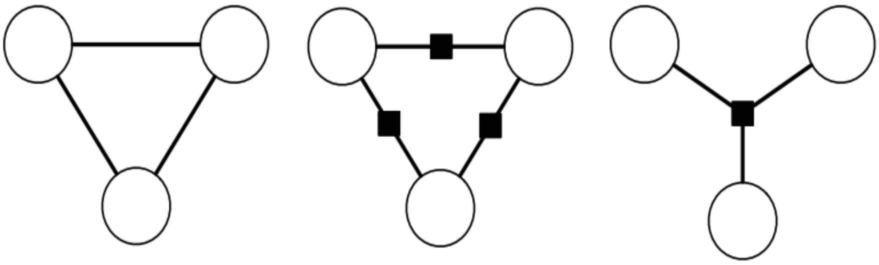

Figure 2: A Markov network with an ambiguous factorization. Both of the factor graphs at right factorize according to the Markov network at left. $p(y_1, y_2, y_3) \sim f(y_1, y_2, y_3)$ for some positive function f, and $p(y_1, y_2, y_3) \sim f(y_1, y_2)g(y_2, y_3)h(y_1, y_3)$. The second model family is a strict subset of the first, so in this smaller family we might not require as much data to accurately estimate the distribution. But Markov network formalism cannot distinguish between these two parameterizations. ???

2.4 Directed Models

A directed graphical model describes how a distribution factorizes into local conditional probability distributions. Let G be a directed acyclic graph, in which $\pi(s)$ are the indices of the parents of Y_s in G. A directed graphical is a family of distributions that factorize as

$$p(\mathbf{y}) = \prod_{s=1}^{S} p(y_s|\mathbf{y}_{\pi(s)}) \tag{4}$$

We refer to the distributions $p(y_s|\mathbf{y}_{\pi(s)})$ as local conditional distributions. Note that $\pi(s)$ can be empty for variables that have no parents, then in this case $p(y_s|\mathbf{y}_{\pi(s)}) = p(y_s)$.

Directed models can be thought of as a kind of factor graph in which the individual factors are locally normalized in a special fashion so that

1. the factors equal conditional distributions over subsets of variables
2. the normalization constant $Z = 1$

Directed models are often used as generative models. Note that, conditional random fields and structured SVM are discrimiative models.

2.5 Problem

The main difference between Markov Random fields and Conditional Random Fields is: there are some observed random variables in CRFs. Given a set of observed random variables X, we would like to predict and output the value of some unobserved output random variables Y. However,we will be interested in building distribution over the combined set of variables $X \cup Y$,

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \prod_{a=1}^{A} \Psi_a(\mathbf{x}_a, \mathbf{y}_a) \tag{5}$$

where now each local function Ψ_a depends on two subsets of variables $X_a \in X$ and $Y_a \in Y$.

2.6 Inference in Graphical Models

Inference is a challenging task, depending on the structure of the graph, and in many case, NP-hard. However, in some cases, the problem is tractable, we can use exact inference algorithms. Otherwise, we have to use approximate inference algorithms.

- Marginal inference: what is the probability of a given variables after sum everything else out (e.g., probability of spam vs non-spam) ?

$$p(y=1) = \sum_{x_1} \sum_{x_2} \cdots \sum_{xn} p(y=1, x_1, ..., x_n) \tag{6}$$

- MAP (Maximum A Posterior) Inference: what is the most likely assignment to the variables, possibly conditioned on evidence (e.g., predicting characters from handwriting).

$$\max_{x_1,\ldots,x_n} p(x_1, ..., x_n | y=1) \tag{7}$$

- Exact inferences:
  - variable elimination
  - message passing
  - junction tree
  - graph cuts
- Approximate inferences:
  - loopy belief propagation
  - linear programming relaxation
  - sampling methods
  - variational methods

3 Generative versus Discriminative Models

Generative models are models that describe how a label vector $\mathbf{y}$ can probabilistically "generate" a feature vector $\mathbf{x}$. Discriminative models work in the reverse direction, describing directly how to take a feature vector $\mathbf{x}$ and assign it a label $\mathbf{y}$. For example, the naive Bayes and HMM are generative, and the logistic regression model is discriminative.

Because a generative model takes the form $p(\mathbf{y}, \mathbf{x}) = p(\mathbf{y})p(\mathbf{x}|\mathbf{y})$, it is often natural to represent a generative model by a directed graph in which outputs $\mathbf{y}$ topologically precede the inputs, which means we first generate $\mathbf{y}$, and then generate $\mathbf{x}$ based on the existing $\mathbf{y}$. Similarly, we will see that it is often natural to represent a discriminative model by a undirected graph. However, this need not always be the case. It can also be useful to depict discriminative models by directed graphs in which the $\mathbf{x}$ precede the $\mathbf{y}$.

3.1 Naive Bayes Classifier

We want to predict a single discrete class variable y given a vector of features $\mathbf{x} = (x_1, ..., x_K)$. One simple way to accomplish this is to **assume** that once the class label is known, all the features are independent. The resulting classifier is called the naive Bayes classifier. It is based on a joint probability model of the form[2]:

$$p(y, \mathbf{x}) = p(y) \prod_{k=1}^{K} p(x_k|y) \tag{8}$$

hence, we can describe this distribution with directed model as below.

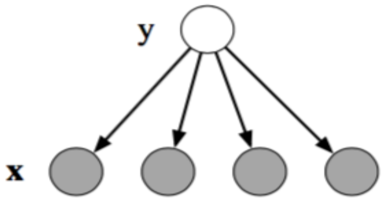

Figure 3: The naive Bayes classifier

We can also write this model as a factor graph, by defining a factor $\Psi(y) = p(y)$, and a factor $\Psi_k(y, x_k) = p(x_k|y)$ for each feature x_k.

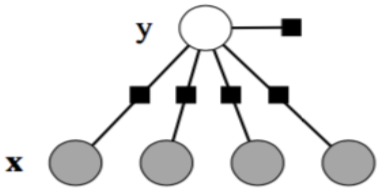

Figure 4: The factor graph of naive Bayes classifier.

3.2 Logistic Regression or Maximum Entropy Classifier

Logistic regression is used in binomial case, but maximum entropy classifier is used in multinomial setting[3]. Both are called log-linear model. They assume that the log probability, $\log p(y|\mathbf{x})$, of each class is a linear function of $\mathbf{x}$, plus a normalization constant. why this assumption ? Thus, we can convert a value into a probability. This leads to the conditional distribution:

$$p(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left\{\theta_y + \sum_{j=1}^{K} \theta_{y,j} x_j\right\} \tag{9}$$

where

$$Z(\mathbf{x}) = \sum_{y} \exp\left\{\theta_y + \sum_{j=1}^{K} \theta_{y,j} x_j\right\} \tag{10}$$

is a normalization constant, and θ_y is a bias weight that acts like $\log p(y)$ in naive Bayes. Note that, y here is a fixed value of a random variable, and $\mathbf{x}$ is the features. We want to know the probability of that data belongs to class y given the K-dimensional data features $\mathbf{x}$.

Feature Functions Now let's define a set of feature functions that are nonzero only for a single class. In other words, each class has one feature function. To do this, the feature functions can be defined as $f_{y',j}(y,\mathbf{x}) = \mathbb{1}_{\{y'=y\}} x_j = \{0 \; or \; x_j\}$ for the feature weight and $f_{y'}(y,\mathbf{x}) = \mathbb{1}_{\{y'=y\}}$ for the bias weights. Now we can use f_k to index each feature function $f_{\{y',j\}}$, and θ_k to index its corresponding weight $\theta_{y',j}$. Using this notation trick, the logistic regression model becomes:

$$p(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left\{\sum_{k=1}^{K} \theta_k f_k(y,\mathbf{x})\right\} \tag{11}$$

Thus, in the summation, it will sum over all data features for the class y. We will introduce feature function in details when we get into the linear-chain CRFs.

3.3 Sequence Models

Classifiers predict only a single class variable, but the true power of graphical model lies in their ability to model many variables that are interdependent. In this section, we discuss perhaps the simplest form of dependency, in which the output variables in the graphical model are arranged in a sequence.

Named-entity Recognition (NER), is the problem of identifying and classifying proper names in text, including locations, such as *China*; people, such as *George Bush*; and organizations, such as the *United Nations*. The system must identify them based only on context. There is a problem if we classify each word independently, for example, the New York and New York Time are different entity. One way to relax this independence assumption is to arrange the output variables in a linear chain, which is taken by hidden Markov model(HMM).

An HMM models a sequences of observation $X = \{x_t\}_{t=1}^{T}$ by assuming that there is an underlying sequence of states $Y = \{y_t\}_{t=1}^{T}$. We let O be the finite set of possible observations and S be a finite set of possible states, i.e. $x_t \in O$ and $y_t \in S$ for all t. To make the joint distribution $p(\mathbf{y}, \mathbf{x})$ tractably, an HMM makes two independence assumption:

1. First, it assumes that each state depends only on its immediate predecessor
2. Second, it also assumes that each observation variable x_t depends only on the current state y_t.

Thus, we can factorize the joint probability as

$$p(\mathbf{y}, \mathbf{x}) = \prod_{t=1}^{T} p(y_t|y_{t-1})p(x_t|y_t) \tag{12}$$

3.4 Comparison: commons

Both generative models and discriminative models describe distribution over $(\mathbf{y}, \mathbf{x})$, but they work in different directions. A generative model, such as the naive Bayes classifier and the HMM, is a family of joint distributions that factorizes as

$$p(\mathbf{y}, \mathbf{x}) = p(\mathbf{y})p(\mathbf{x}|\mathbf{y}) \tag{13}$$

that is, it describes how to sample, or "generate", values for features given the label. A discriminative model, such as the logistic regression model, is a family of conditional distribution $p(\mathbf{y}|\mathbf{x})$, that is, the classification rule is modeled directly. In principle, a discriminative model could also be used to obtain a joint distribution $p(\mathbf{y}, \mathbf{x})$ by supplying a marginal distribution $p(\mathbf{x})$ over the inputs, but this is rarely needed.

In addition, to include interdependent features in a generative model, we have two choice:

1. The first choice is to enhance the model to represent dependencies among the inputs, e.g., by adding directed edges among each $\mathbf{x}_t$. But this is often difficult to do while retaining tractability.
2. The second choice is to make simplifying independence assumptions, such as the naive Bayes assumption. This idea can sometimes work well. But it can also be problematic because the independence assumptions can hurt performance.

3.5 Comparison: differences

The main difference between discriminative and generative models is that a conditional distribution $p(\mathbf{y}|\mathbf{x})$ does not include a model of $p(\mathbf{x})$, which is not needed for classification anyway. **The difficulty in modeling $p(\mathbf{x})$ is that it often contains many highly dependent features that are difficult to model. By modeling the the conditional distribution directly, we can remain agnostic about the form of $p(\mathbf{x})$.** For example, the family of naive Bayes distribution has conditionals in the "logistic regression form". But there are many other joint models, some with complex dependencies among $\mathbf{x}$, whose conditional distributions also have the form (10). They have totally different $p(\mathbf{x})$.

The principal advantage of discriminative modeling, such as CRFs, is they make conditional independence assumptions among $\mathbf{y}$, and assumptions about how the $\mathbf{y}$ can depend on $\mathbf{x}$, but do not make conditional independence assumptions among $\mathbf{x}$. This point also be understood graphically.

Suppose that we have a factor graph representation for the joint distribution $p(\mathbf{y}, \mathbf{x})$. The random variables in the left of bipartite graph can be a set of input variables X that we assume are always observed, and a set of output variables Y that we wish to predict. The factors lies on the right of bipartite graph. We will be interested in building distributions over the combined set of variable $X \cup Y$, and the undirected model over X and Y would be given by

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \prod_{a=1}^{A} \Psi_a(\mathbf{x}_a, \mathbf{y}_a) \tag{14}$$

where now each local function Ψ_a depends on two subsets of variables $X_a \subseteq X$ and $Y_a \subseteq Y$. The normalization constant becomes

$$Z = \sum_{\mathbf{x},\mathbf{y}} \prod_{a \in F} \Psi_a(\mathbf{x}_a, \mathbf{y}_a) \tag{15}$$

which now involves summing over all assignments both to $\mathbf{x}$ and $\mathbf{y}$. If we want to construct a factor graph for the conditional distribution $p(\mathbf{y}|\mathbf{x})$, any factors that depend only on $\mathbf{x}$ vanish from the graphical structure for the conditional distribution. They are irrelevant to the conditional because they are constant with respect to $\mathbf{y}$

3.6 Naive Bayes and Logistic Regression Form a Generative-Discriminative Pair

The difference between naive Bayes and logistic regression is due only to the fact that the first is generative and the second discriminative. That is, naive Bayes use the $p(y, \mathbf{x})$ but logistic regression use $p(y|\mathbf{x})$ for classification. The two classifiers are, for discrete input, identical in all other respect. The navie Bayes model defines the same family of distributions as the logistic regression model, if we interpret the logistic regression generatively as

$$p(y, \mathbf{x}) = \frac{\exp\left\{\sum_{k=1}^{K} \theta_k f_k(y, \mathbf{x})\right\}}{\sum_{\hat{y},\hat{\mathbf{x}}} \exp\left\{\sum_{k=1}^{K} \theta_k f_k(\hat{y}, \hat{\mathbf{x}})\right\}} \tag{16}$$

This means that if the navie Bayes model is trained to maximize the conditional likelihood, we recover the same classifier as from logistic regression. Then, we have

$$p(y|\mathbf{x}) = \frac{p(y, \mathbf{x})}{p(\mathbf{x})} = \frac{p(y) \prod_{k=1}^{K} p(x_k|y)}{\sum_y p(y) \prod_{k=1}^{K} p(x_k|y)} \tag{17}$$

and we get the optimal y^* by using MAP

$$y^* = \arg\max_y p(y|\mathbf{x}) = \arg\max_y p(y, \mathbf{x}) = \arg\max_y p(y) \prod_{k=1}^{K} p(x_k|y) \tag{18}$$

Conversely, if the logistic regression model is interpreted generatively, as in (16), and is trained to maximize the joint likelihood $p(y, \mathbf{x})$, then we recover the same classifier as from naive Bayes.

3.7 Why different ?

So far, we may still not clear why these approaches should be so different, because we can always convert between the two methods using Bayes rule. Generative and discriminative models both have the aim of estimating $p(\mathbf{y}|\mathbf{x})$, but they get there in different ways, or they get different estimate in practice. Estimating $p(\mathbf{y})p(\mathbf{x}|\mathbf{y})$ first, and then computing the resulting $p(\mathbf{y}|\mathbf{x})$ (the generative approach) yields a different estimate than estimating $p(\mathbf{y}|\mathbf{x})$ directly.

If we managed to obtain the "true" underlying distribution from which our data were actually sampled, that is

$$p^*(\mathbf{y}, \mathbf{x}) = p^*(\mathbf{y})p(\mathbf{x}|\mathbf{y}), \tag{19}$$

then we can simply compute the true $p^*(\mathbf{y}|\mathbf{x})$, which is exactly the target of the discriminative approach. But we never have the "true" distribution.

Insight of Generative and Discriminative Model.
The fundamental different between discriminative models and generative models is[4]:

- Discriminative model learn the (hard or soft) boundary between classes
- Generative models model the distribution of individual classes

Also, there is an interpretation from the perspective of degree of freedom and overfitting.

3.8 Some Advantages of Generative Models

1. Generative models can be more natural for handling latent variables, partially-labeled data, and unlabeled data. In the most extreme case, when the data is entirely unlabeled, generative models can be applied in an unsupervised fashion.
2. In some cases, a generative model can perform better than a discriminative model, intuitively because the input model $p(\mathbf{x})$ may have a smoothing effect on the conditional.
3. Sometimes either the problem suggests a natural generative model, or the application requires the ability to predict both future inputs and future outputs, making a generative model preferable.

References

[1] Sutton, Charles, and Andrew McCallum. "An introduction to conditional random fields." Foundations and Trends® in Machine Learning 4.4 (2012): 267-373.

[2] https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[3] https://www.quora.com/What-is-the-relationship-between-Log-Linear-model-MaxEnt-/model-and-Logistic-Regression

[4] https://stats.stackexchange.com/questions/12421/generative-vs-discriminative

[5] http://pages.cs.wisc.edu/~jerryzhu/cs769/CRF.pdf

[6] Log-Linear Model in Markov Network https://homes.cs.washington.edu/~nasmith/papers/smith.tut04.pdf