Pattern Recognition Advanced

Discriminative Graphical Models: Conditional Random Fields

mcuturi@i.kyoto-u.ac.jp

Today's talk

- Seen recently: hidden markov models, latent variables

- Today, present **Conditional Random Fields** (ICML 2001).

  Conditional random fields: Probabilistic models for segmenting and labeling sequence data, by Lafferty McCallum Pereira

- Proposed by the authors when working for (now defunct) WhizBang! labs.

- WhizBang! labs was a company specialized in extracting automatically information from web-pages.

- Objective: parse millions of webpages to select important content

  - job advertisements
  - company reports

- Problem: recover structure in very large databases.

Reference text: An Introduction to Conditional Random Fields Sutton, McCallum

Today's talk

Objective :**Annotate Subparts of Large Complex Objects**

- The theory is a general and applies to "random fields".
- Difference with Hidden Markov Models: **we do not use** a **generative model**

$$\boldsymbol{X} = \text{cat eat mice}, \qquad \mathbf{Y} = \mathbf{N\ V\ N}$$

$$P(\underbrace{\boldsymbol{X}}_{\text{text}}, \underbrace{\boldsymbol{Y}}_{\text{parsing result}})$$

- But only a **discriminative** approach, $i.e.$ we only focus on

$$P(Y|X)$$

- Difference? $\mathbf{P}(\mathbf{X}, \mathbf{Y}) = P(Y|X)\mathbf{P}(\mathbf{X})$. **no need to take care of** $P(X)$.

Graphical Models

an introduction

Structured Predictions

- For many applications, predicting **many joint variables** is fundamental.
- Examples
  - classify regions of an image,
  - segmenting genes in a strand of DNA,
  - extract syntax from natural-language text
- The goal is to **produce local predictors**

$$\mathbf{y} = \{y_0, y_1, \ldots, y_T\} \text{ given } \mathbf{x}$$

- Of course, one could only focus on individual regression/classification task

$$\mathbf{x} \mapsto y_s, \text{ for each } s,$$

independently... but then how can we make sure the final answer is **coherent**?

Graphical Models

- A natural way to model constraints on output variables is provided by graphical models, *e.g.*
  - Bayesian networks,
  - Neural networks,
  - factor graphs,
  - Markov random fields,
  - Ising models, *etc.*
- **Graphical models** represent a complex distribution over many variables as a product of **local** *factors* on smaller subsets of variables.
- Two types of graphical models: **directed** and **undirected**

Some Notations First

- We consider probabilities on variables **indexed** by $V = X \cup Y$,
  - X is a set of **input variables**
  - Y is a set of **output variables** that we wish to predict.
- We assume that each variable takes values in a **discrete set**.
- An assignment to all variables indexed in X (resp. Y) is denoted $\mathbf{x}$ (resp. $\mathbf{y}$).
- An assignment to all variables indexed in X and Y is denoted $\mathbf{z} = (\mathbf{x}, \mathbf{y})$.
  - For $s \in X$, x_s denotes the value assigned to s by $\mathbf{x}$.
  - For $s \in Y$, y_s denotes the value assigned to s by $\mathbf{y}$.
  - For $v \in V$, z_s denotes the value assigned to s by $\mathbf{z}$.
  - For a subset $a \subset V$, $\mathbf{z}_a = (z_s)_{s \in a}$.

Undirected Graphical Models

- Given a collection of subsets $\mathcal{F} \subset \wp(V)$, an **undirected graphical *model*** is the **set of all distributions** that can be written as

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \prod_{a \in \mathcal{F}} \Psi_a(\mathbf{z}_a),$$

for any choice of *local function* $F = \{\Psi_a\}$, where $\Psi_a : \mathcal{V}^{|a|} \to \mathbb{R}_+$.

Undirected Graphical Models

$$\boxed{p(\mathbf{x},\mathbf{y}) = \frac{1}{Z}\prod_{a\in\mathcal{F}}\Psi_a(\mathbf{z}_a)}$$

- Usually sets a are much smaller than the full variable set V.
- Z is a normalization factor, defined as

$$Z = \sum_{\mathbf{x},\mathbf{y}}\prod_{a\in\mathcal{F}}\Psi_a(\mathbf{z}_a).$$

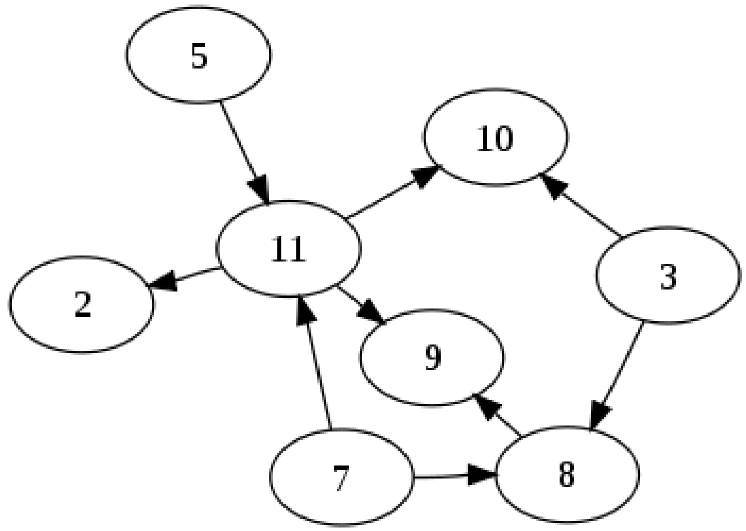
- Computations are easier if each local function is an exponential model:

$$\Psi_a(\mathbf{x}_a,\mathbf{y}_a) = \exp\left\{\sum_k \theta_{ak} f_{ak}(\mathbf{z}_a)\right\},$$

- For each k and subset of variables a, a **weighted** feature $f_{ak}(\mathbf{z}_a)$ with $\boldsymbol{\theta_{ak}}$.

Directed Graphical Model

- Let $G = (V, E)$ be a **directed** acyclic graph.
- For each v, $\pi(v) \subset V$ is the set of parents of v in G.



- A **directed** graphical model is a family of distributions that factorize as:

$$p(\mathbf{y}, \mathbf{x}) = \prod_{v \in V} p(z_v | \mathbf{z}_{\pi(v)}).$$

- Difference: not only subsets a, but also directions, given by π.

Starting Slowly: Naive Bayes

Text Classes

- Suppose a whole text can only belong to **one** category.

$$\text{TEXT} \overset{?}{\leftrightarrow} \text{CATEGORY}$$

- Here, we assume also that there is a **joint** probability on texts and their category.

$$P(\texttt{text}, \texttt{category})$$

which quantifies how likely the match between

a text `text` and a category `category` is

- For instance,

$$P(\texttt{'I am feeling hungry these days'}, \texttt{'poetry'}) \approx 0$$

$$P(\texttt{'Manchester United's stock rose after their victory'}, \texttt{'business'})$$
$$\vee$$
$$P(\texttt{'Manchester United's stock rose after their victory'}, \texttt{'sports'})$$

Text classification & probabilistic framework

- Hence, given a sequence of words (including punctuation),

$$\mathbf{w} = (w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, \cdots, w_n)$$

- assuming we know P, the **joint** probability between texts and categories,
- an easy way to guess the category of $\mathbf{w}$ is by looking at

$$\text{category-prediction}(\mathbf{w}) = \operatorname*{argmax}_{\boldsymbol{C}} P(\boldsymbol{C} | \boldsymbol{w_1, w_2, \cdots, w_n})$$

Text classification & probabilistic framework

$$P(\texttt{'poetry'} | \texttt{`I am feeling hungry these days'}) = 0.0037$$

$$P(\texttt{'business'} | \texttt{`I am feeling hungry these days'}) = 0.005$$

$$P(\texttt{'sports'} | \texttt{`I am feeling hungry these days'}) = 0.003$$

$$P(\texttt{'food'} | \texttt{`I am feeling hungry these days'}) = 0.2$$

$$P(\texttt{'economy'} | \texttt{`I am feeling hungry these days'}) = 0.04$$

$$P(\texttt{'society'} | \texttt{`I am feeling hungry these days'}) = 0.08$$

Text classification & probabilistic framework

$$P(\texttt{'poetry'}|\texttt{`I am feeling hungry these days'}) = 0.0037$$
$$P(\texttt{'business'}|\texttt{`I am feeling hungry these days'}) = 0.005$$
$$P(\texttt{'sports'}|\texttt{`I am feeling hungry these days'}) = 0.003$$
$$\rightarrow P(\texttt{'food'}|\texttt{`I am feeling hungry these days'}) = 0.2$$
$$P(\texttt{'economy'}|\texttt{`I am feeling hungry these days'}) = 0.04$$
$$P(\texttt{'society'}|\texttt{`I am feeling hungry these days'}) = 0.08$$

Bayes Rule

- Using Bayes theorem $p(A,B) = p(A|B)p(B)$,

$$P(\boldsymbol{C}|\boldsymbol{w_1},\boldsymbol{w_2},\cdots,\boldsymbol{w_n}) = \frac{P(\boldsymbol{C},\boldsymbol{w_1},\boldsymbol{w_2},\cdots,\boldsymbol{w_n})}{P(\boldsymbol{w_1},\boldsymbol{w_2},\cdots,\boldsymbol{w_n})}$$

- When looking for the category C that best fits $\mathbf{w}$, we only focus on the numerator.
- Bayes theorem also gives that

$$\begin{aligned} P(\boldsymbol{C},\boldsymbol{w_1},\cdots,\boldsymbol{w_n}) &= P(\boldsymbol{C})P(\boldsymbol{w_1},\boldsymbol{w_2},\cdots,\boldsymbol{w_n}|\boldsymbol{C}) \\ &= P(\boldsymbol{C})P(\boldsymbol{w_1}|\boldsymbol{C})P(\boldsymbol{w_2},\boldsymbol{w_3},\cdots,\boldsymbol{w_n}|\boldsymbol{C},\boldsymbol{w_1}) \\ &= P(\boldsymbol{C})P(\boldsymbol{w_1}|\boldsymbol{C})P(\boldsymbol{w_2}|\boldsymbol{C},\boldsymbol{w_1})P(\boldsymbol{w_3},\boldsymbol{w_4},\cdots,\boldsymbol{w_n}|\boldsymbol{C},\boldsymbol{w_1},\boldsymbol{w_2}) \\ &= P(\boldsymbol{C})\prod_{i=1}^{n} P(\boldsymbol{w_i}|\boldsymbol{C},\boldsymbol{w_1},\cdots,\boldsymbol{w_{i-1}}) \end{aligned}$$

Examples

- Assume we have the beginning of this news title

$$w_1, \cdots, w_{12} = \text{`The weather was so bad that the organizers decided to close the'}$$

- If C =business, then

$$P(W_{13} = \text{`market'} | \text{business}, w_1, \cdots, w_{12})$$

should be quite high, as well as summit, meeting *etc.*.

- On the other hand, if we know C =sports, the probability for w_{13} changes significantly...

$$P(W_{13} = \text{`game'} | \text{sports}, w_1, \cdots, w_{12})$$

The Naive Bayes Assumption

- From a factorization

$$P(C, w_1, \cdots, w_n) = P(C) \prod_{i=1}^{n} P(w_i | C, w_1, \cdots, w_{i-1})$$

which handles all the **conditional** structures of text,

- we assume that each word appears **independently conditionally to** C,

$$\begin{aligned} P(w_i | C, w_1, \cdots, w_{i-1}) &= P(w_i | C, \cancel{w_1, \cdots, w_{i-1}}) \\ &= P(w_i | C) \end{aligned}$$

- and thus

$$P(C, w_1, \cdots, w_n) = P(C) \prod_{i=1}^{n} P(w_i | C)$$

Naive Bayes & Logistic Regression
Binary Case

Naive Bayes

Recall the **Naive Bayes Assumption** on $p(\mathbf{x}, y)$

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

- Bayes classifier can be interpreted as a **directed** graphical model, where
  - $V = \{X = \{1, \cdots, N\}\} \cup \{Y = \mathbf{1}\}$
  - All elements of X have only one parent:

$$\pi(i) = \mathbf{1}.$$

Logistic Regression

- Famous technique for classification (with binary variables):

Logistic Regression (or Maximum Entropy Classifier), model $p(y|\mathbf{x})$

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

- by malaxing things a bit, introducing
  - $f_{y',j}(y,\mathbf{x}) = \delta_{y'=y}\, x_j$
  - $f_{y'}(y,\mathbf{x}) = \delta_{y'=y}$
- and renumbering all these functions (and the corresponding weights $\theta_{y,j}$ and θ_y) 1 to K,

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

we obtain an **undirected** graphical model.

A Simple Example: Classification

Naive Bayes Assumption, $p(\mathbf{x}, y)$

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

equivalent to a **directed** graphical model

Logistic Regression, $p(y|\mathbf{x})$

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

equivalent to an **undirected** graphical model

Link between Naive Bayes and Logistic Regression

Deriving the conditional distribution $p(y|\mathbf{x})$ of **Naive Bayes**

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

- Let us study the case where **all** variables are binary.

Link between Naive Bayes and Logistic Regression

- Set

$$p_1 = P(y = 1)$$
$$p_{i0} = P(x_i = 1 | y = 0)$$
$$p_{i1} = P(x_i = 1 | y = 1)$$

- Then

$$p(\mathbf{x}_i = x_i | \mathbf{y} = y) = p_{i0}^{(1-y)x_i}(1 - p_{i0})^{(1-y)(1-x_i)} p_{i1}^{y x_i}(1 - p_{i1})^{y(1-x_i)}$$

and

$$p(\mathbf{y} = y) = p_1^y (1 - p_1)^{1-y}$$

- Define

$$\theta_0 = \log \frac{p_1}{1 - p_1} + \sum_{i=1}^{n} \log \frac{1 - p_{i1}}{1 - p_{i0}}$$
$$\phi_i = \log \frac{p_{i0}}{1 - p_{i0}}$$
$$\theta_i = \log \frac{1 - p_{i0}}{p_{i0}} \frac{p_{i1}}{1 - p_{i1}}$$

Source: Y.Bulatov

Link between Naive Bayes and Logistic Regression

- then

$$p(\mathbf{x}, y) = \frac{e^{\theta_0 y}\, e^{\sum_{i=1}^{N} \phi_i x_i}\, e^{\sum_{i=1}^{N} \theta_i y x_i}}{\prod_{i=1}^{N}(1 + e^{\phi_i}) + e^{\theta_0} \prod_{i=1}^{N}(1 + e^{\theta_i + \phi_i})}$$

- which can be decomposed again as

$$\begin{aligned} p(\mathbf{x}, y) &= \frac{e^{\left(\theta_0 + \sum_{i=1}^{N} \theta_i x_i\right)^y}}{1 + e^{\theta_0 + \sum_{i=1}^{N} \theta_i x_i}} \times \frac{e^{\sum_{i=1}^{N} \phi_i x_i}\left(1 + e^{\theta_0 + \sum_{i=1}^{N} \theta_i x_i}\right)}{\prod_{i=1}^{N}(1 + e^{\phi_i}) + e^{\theta_0} \prod_{i=1}^{N}(1 + e^{\theta_i + \phi_i})} \\ &= p(y|\mathbf{x}) \quad \times \quad p(\mathbf{x}) \end{aligned}$$

- We have highlighted the conditional distribution induced by naive Bayes in the case of binary variables.
- This conditional distribution coincides with the logistic regression form
- This can be shown for many other cases ($e.g.$ $p(x_k|y)$ is Gaussian)

Next Example, Sequence Models

Predict the corresponding structure $Y = 1, \cdots, T$ of T words, $X = 1, \cdots, T$

Recall the **Hidden Markov Model** on $p(\mathbf{x}, \mathbf{y})$

$$p(\mathbf{x}, \mathbf{y}) = p(y_1) \prod_{k=1}^{N} p(y_t | y_{t-1}) p(x_t | y_t)$$

- Of course, HMM's are **directed** graphical model, where
  - $V = \{X = \{1, \cdots, T\}\} \cup \{Y = \{\mathbf{1}, \cdots, \mathbf{T}\}\}$
  - Each element of X has only one parent:

$$\pi(i) = \mathbf{i}.$$

  - Each element of $\{\mathbf{2}, \cdots, \mathbf{T}\}$ has one parent:

$$\pi(\mathbf{i}) = \mathbf{i} - \mathbf{1}.$$

Sequence Models

The **Linear Conditional Random Field** on $p(\mathbf{y}|\mathbf{x})$

- A *linear-chain CRF* is a distribution $p(\mathbf{y}|\mathbf{x})$ that takes the form

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \exp\left\{ \sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right\},$$

  where $Z(\mathbf{x})$ is an instance-specific normalization function

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{t=1}^{T} \exp\left\{ \sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right\}.$$

- The Linear-Chain CRF is an **undirected** graphical model

From HMM to Linear CRF

- Let us rewrite the HMM density

$$p(\mathbf{y},\mathbf{x}) = \frac{1}{Z}\prod_{t=1}^{T}\exp\left\{\sum_{i,j\in S}\theta_{ij}\mathbf{1}_{\{y_t=i\}}\mathbf{1}_{\{y_{t-1}=j\}} + \sum_{i\in S}\sum_{o\in O}\mu_{oi}\mathbf{1}_{\{y_t=i\}}\mathbf{1}_{\{x_t=o\}}\right\},$$

  where S (states) is the set of values possibly taken by y and O (outputs) by x.

- Every HMM can be written in this form by setting

$$\theta_{ij} = \log p(y' = i|y = j) \text{ and } \mu_{oi} = \log p(x = o|y = i).$$

From HMM to Linear CRF

- We can highlight again the **feature functions** perspective:
- Each feature function has the form

$$f_k(y_t, y_{t-1}, x_t).$$

- There needs to be one feature for each **transition** (i, j),

$$f_{ij}(y, y', x) = \mathbf{1}_{\{y=i\}}\mathbf{1}_{\{y'=j\}}$$

and one feature for each **state-observation pair** (i, o),

$$f_{io}(y, y', x) = \mathbf{1}_{\{y=i\}}\mathbf{1}_{\{x=o\}}$$

- Once this is done, we get

$$p(\mathbf{y}, \mathbf{x}) = \frac{1}{Z}\prod_{t=1}^{T}\exp\left\{\sum_{k=1}^{K}\theta_k f_k(y_t, y_{t-1}, x_t)\right\}.$$

where f_k ranges over both all of the f_{ij} and all of the f_{io}.

From HMM to Linear CRF

- Last step: write the conditional distribution $p(\mathbf{y}|\mathbf{x})$ induced by HMM's

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{y},\mathbf{x})}{\sum_{\mathbf{y}'} p(\mathbf{y}',\mathbf{x})} = \frac{\prod_{t=1}^{T} \exp\left\{\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t)\right\}}{\sum_{\mathbf{y}'} \prod_{t=1}^{T} \exp\left\{\sum_{k=1}^{K} \theta_k f_k(y'_t, y'_{t-1}, x_t)\right\}}.$$

- this is the linear CRF induced by HMM's...

Differences between HMM and Linear CRF

- If $p(\mathbf{y}, \mathbf{x})$ factorizes as an HMM $\Rightarrow$ distribution $p(\mathbf{y}|\mathbf{x})$ is a linear-chain CRF.

However, other types of linear-chain CRFs,
not induced by HMM's,
are also useful

- For example,
  - in an HMM, a transition from state i to j receives the same score,

$$\log p(y_t = j | y_{t-1} = i),$$

    regardless of the x_{t-1}.
  - In a CRF, the score of the transition (i, j) might depend **for instance** on the current observation vector, $e.g.$ by defining

$$f_k = \mathbf{1}_{\{y_t = j\}} \mathbf{1}_{\{y_{t-1} = 1\}} \mathbf{1}_{\{x_t = o\}}.$$

General CRF

$p(\mathbf{y}|\mathbf{x})$ is a conditional random field
if the distribution $p(\mathbf{y}|\mathbf{x})$ can be written as
$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{\Psi_a \in \mathcal{F}} \exp\left\{\sum_{k=1}^{K(a)} \theta_{ak} f_{ak}(\mathbf{y}_a, \mathbf{x}_a)\right\}.$

- Many parameters potentially...
- For linear chain CRF, same weights/functions are used for factors $\Psi_t(y_t, y_{t-1}, \mathbf{x}_t)$, $\forall t$.
- **Solution**: Partition set of subsets of variables $\mathcal{F}$ into groups $\mathcal{F} = \mathcal{F}_1, \cdots, \mathcal{F}_P$.
- Each subset $\mathcal{F}_i$ is a set of subsets of variables which share the same local functions, $i.e.$

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{\mathcal{F}_i \in \mathcal{F}} \prod_{\Psi_a \in \mathcal{F}_i} \Psi_a(\mathbf{y}_a, \mathbf{x}_a)$$

  where

$$\Psi_a(\mathbf{y}_a, \mathbf{x}_a) = \exp\left\{\sum_{k=1}^{K(\boldsymbol{i})} \theta_{\boldsymbol{i}k} f_{\boldsymbol{i}k}(\mathbf{y}_a, \mathbf{x}_a)\right\}.$$

- Most CRF's of interest implement such structures.

Features - Factorization

- CRF's are very general **structures**. What about the practical implementation?
- Features depend on the task. In some NLP tasks with linear CRF,

$$f_{pk}(\mathbf{y}_c, \mathbf{x}_c) = \mathbf{1}_{\{\mathbf{y}_c = \tilde{\mathbf{y}}_c\}} q_{pk}(\mathbf{x}_c).$$

- Each feature is **factorized**
  - is nonzero only for a single output configuration $\tilde{\mathbf{y}}_c$,
  - its value only depends input observation $\mathbf{x}_c$.
- This **factorization** is attractive because computationally efficient:
  - computing each q_{pk} may involve nontrivial text or image processing,
  - However, we only need to evaluate it **once**, even if it shared across many features.
- These functions $q_{pk}(\mathbf{x}_c)$ are called **observation functions**.
- Examples of observation functions are
  - "word x_t is capitalized",
  - "word x_t ends in *ing*".

Learning with Linear Chain CRF's

Estimation and Prediction

A *linear-chain CRF* is a distribution $p(\mathbf{y}|\mathbf{x})$ that takes the form

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \exp\left\{ \sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right\},$$

- Two major tasks ahead:

Given a set of features f_k, estimate all parameters θ_k

Predict the labels of a new input $\mathbf{x}$, $\mathbf{y}^* = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})$.

- We first review the **prediction** task, **estimation** is covered next.
- In the **prediction** task, we will re-use the **Forward-Backward and Viterbi algorithms** of HMM's.

Prediction - Backward Forward

- The HMM's distribution can be factorized as a directed graphical model

$$p(\mathbf{y},\mathbf{x}) = \prod_t \Psi_t(y_t, y_{t-1}, x_t)$$

(with $Z = 1$) and factors defined as:

$$\Psi_t(j,i,x) \stackrel{\text{def}}{=} p(y_t = j | y_{t-1} = i) p(x_t = x | y_t = j).$$

- The HMM forward algorithm, used to compute the probability $p(\mathbf{x})$ of observations, uses the summation.

$$\begin{aligned} p(\mathbf{x}) &= \sum_{\mathbf{y}} p(\mathbf{x},\mathbf{y}) = \sum_{\mathbf{y}} \prod_{t=1}^{T} \Psi_t(y_t, y_{t-1}, x_t) \\ &= \sum_{y_{\mathrm{T}}} \sum_{y_{\mathrm{T}-1}} \Psi_{\mathrm{T}}(y_{\mathrm{T}}, y_{\mathrm{T}-1}, x_{\mathrm{T}}) \sum_{y_{\mathrm{T}-2}} \Psi_{\mathrm{T}-1}(y_{\mathrm{T}-1}, y_{\mathrm{T}-2}, x_{\mathrm{T}-1}) \sum_{y_{\mathrm{T}-3}} \cdots \end{aligned}$$

- Idea: **cache** intermediate sum which are reused **many times** during the computation of the outer sum.

Prediction - Forward

- In that sense, define **forward variables** $\alpha_t \in \mathbb{R}^M$ (where M is the number of states),

$$\begin{aligned}\alpha_t(j) &\stackrel{\text{def}}{=} p(\mathbf{x}_{\langle 1\ldots t\rangle}, y_t = j)\\ &= \sum_{\mathbf{y}_{\langle 1\ldots t-1\rangle}} \Psi_t(j, y_{t-1}, x_t) \prod_{t'=1}^{t-1} \Psi_{t'}(y_{t'}, y_{t'-1}, x_{t'}),\end{aligned}$$

- The summation over $\mathbf{y}_{\langle 1\ldots t-1\rangle}$ ranges over **all** assignments to $y_1, y_2, \ldots, y_{t-1}$.
- The α_t can be computed by the recursion

$$\alpha_t(j) = \sum_{i\in S} \Psi_t(j, i, x_t)\alpha_{t-1}(i),$$

  with initialization $\alpha_1(j) = \Psi_1(j, y_0, x_1)$. (Recall that y_0 is the fixed initial state of the HMM.)
- We can check that $p(\mathbf{x}) = \sum_{y_{\mathrm{T}}} \alpha_{\mathrm{T}}(y_{\mathrm{T}})$.

Prediction - Backward

- Define a **backward recursion**, with reverse order: introduce β_t's

$$\beta_t(i) \stackrel{\text{def}}{=} p(\mathbf{x}_{\langle t+1\ldots \mathrm{T}\rangle} | y_t = i)$$
$$= \sum_{\mathbf{y}_{\langle t+1\ldots \mathrm{T}\rangle}} \prod_{t'=t+1}^{\mathrm{T}} \Psi_{t'}(y_{t'}, y_{t'-1}, x_{t'}),$$

  and the recursion

$$\beta_t(i) = \sum_{j \in S} \Psi_{t+1}(j, i, x_{t+1}) \beta_{t+1}(j),$$

- Initialization: $\beta_{\mathrm{T}}(i) = 1$.
- Analogously to the forward case, $p(\mathbf{x})$ can be computed using the backward variables as

$$p(\mathbf{x}) = \beta_0(y_0) \stackrel{\text{def}}{=} \sum_{y_1} \Psi_1(y_1, y_0, x_1) \beta_1(y_1).$$

Prediction - Forward Backward

- The FB recursions can be combined to obtain the marginal distributions

$$p(y_{t-1}, y_t|\mathbf{x})$$

- Two **perspectives** can be applied, with identical result:
- Taking first a **probabilistic** viewpoint we can write

$$\begin{aligned} p(y_{t-1}, y_t|\mathbf{x}) &= \frac{p(\mathbf{x}|y_{t-1}, y_t)p(y_t, y_{t-1})}{p(\mathbf{x})} \\ &= \frac{p(\mathbf{x}_{\langle 1\ldots t-1\rangle}, y_{t-1})p(y_t|y_{t-1})p(x_t|y_t)p(\mathbf{x}_{\langle t+1\ldots \mathrm{T}\rangle}|y_t)}{p(\mathbf{x})} \\ &\propto \alpha_{t-1}(y_{t-1})\Psi_t(y_t, y_{t-1}, x_t)\beta_t(y_t), \end{aligned}$$

where in the second line we have used the fact that $\mathbf{x}_{\langle 1\ldots t-1\rangle}$ is independent from $\mathbf{x}_{\langle t+1\ldots \mathrm{T}\rangle}$ and from x_t given y_{t-1}, y_t.

Prediction - Forward Backward

- Taking a **factorization** perspective, we see that

$$p(y_{t-1}, y_t, \mathbf{x}) = \Psi_t(y_t, y_{t-1}, x_t) \left(\sum_{\mathbf{y}_{\langle 1 \ldots t-2 \rangle}} \prod_{t'=1}^{t-1} \Psi_{t'}(y_{t'}, y_{t'-1}, x_{t'}) \right) \left(\sum_{\mathbf{y}_{\langle t+1 \ldots \mathrm{T} \rangle}} \prod_{t'=t+1}^{T} \Psi_{t'}(y_{t'}, y_{t'-1}, x_{t'}) \right),$$

which can be computed from the forward and backward recursions as

$$p(y_{t-1}, y_t, \mathbf{x}) = \alpha_{t-1}(y_{t-1}) \Psi_t(y_t, y_{t-1}, x_t) \beta_t(y_t).$$

- With $p(y_{t-1}, y_t, \mathbf{x})$, renormalize over y_t, y_{t-1} to obtain the desired marginal $p(y_{t-1}, y_t | \mathbf{x})$.

Prediction - Forward Backward

- To compute the **globally most probable assignment** $\mathbf{y}^* = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})$,
- we observe that the trick earlier still works if all summations are replaced by maximization.
- This yields the Viterbi recursion:

$$\delta_t(j) = \max_{i \in S} \Psi_t(j, i, x_t)\delta_{t-1}(i)$$

Prediction - Forward Backward in Linear CRF's

- Natural **generalization** of forward-backward and Viterbi algorithms to linear-chain CRFs
- Only transition weights $\Psi_t(j, i, x_t)$ need to be redefined.
- The CRF model can be rewritten as:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \Psi_t(y_t, y_{t-1}, \mathbf{x}_t),$$

where we define

$$\Psi_t(y_t, y_{t-1}, \mathbf{x}_t) = \exp\left\{\sum_k \theta_k f_k(y_t, y_{t-1}, \mathbf{x}_t)\right\}.$$

- Using these definitions, use identical algorithms.
- Instead of computing $p(\mathbf{x})$ as in an HMM, in a CRF the forward and backward recursions compute $Z(\mathbf{x})$.

Parameter Estimation

- Suppose we have i.i.d training data

$$\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{N},$$

  - each $\mathbf{x}^{(i)} = \{\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \ldots \mathbf{x}_T^{(i)}\}$ is a sequence of inputs,
  - each $\mathbf{y}^{(i)} = \{y_1^{(i)}, y_2^{(i)}, \ldots y_T^{(i)}\}$ is a sequence of the desired predictions.

- Parameter estimation can be performed by **penalized maximum conditional likelihood**.

$$\ell(\theta) = \sum_{i=1}^{N} \log p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}).$$

namely,

$$\ell(\theta) = \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=1}^{K} \theta_k f_k(y_t^{(i)}, y_{t-1}^{(i)}, \mathbf{x}_t^{(i)}) - \sum_{i=1}^{N} \log Z(\mathbf{x}^{(i)})$$