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).

  - 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](https://www.cs.cmu.edu/~jav/sutton-crf.html) by Sutton, McCallum

PRA - 2013
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**

\[ X = \text{cat eat mice}, \quad Y = N \triangledown N \]

\[ P(\underbrace{X}_{\text{text}}, \underbrace{Y}_{\text{parsing result}}) \]

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

\[ P(Y|X) \]

- Difference? \[ P(X, Y) = P(Y|X)P(X). \] **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**

  \[ y = \{y_0, y_1, \ldots, y_T\} \text{ given } x \]

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

  \[ x \mapsto y_s, \text{ for each } s, \]

  independently... but then how can we make sure the final answer is **coherent**?
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 $x$ (resp. $y$).
- An assignment to all variables indexed in $X$ and $Y$ is denoted $z = (x, y)$.
  - For $s \in X$, $x_s$ denotes the value assigned to $s$ by $x$.
  - For $s \in Y$, $y_s$ denotes the value assigned to $s$ by $y$.
  - For $v \in V$, $z_s$ denotes the value assigned to $s$ by $z$.
  - For a subset $a \subseteq V$, $z_a = (z_s)_{s \in a}$.
Given a collection of subsets $\mathcal{F} \subset \mathcal{P}(V)$, an undirected graphical model is the set of all distributions that can be written as

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

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

**Undirected Graphical Models**
Undirected Graphical Models

\[ p(x, y) = \frac{1}{Z} \prod_{a \in F} \Psi_a(z_a) \]

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

\[
Z = \sum_{x, y} \prod_{a \in F} \Psi_a(z_a).
\]

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

\[
\Psi_a(x_a, y_a) = \exp \left( \sum_k \theta_{ak} f_{ak}(z_a) \right),
\]

- For each \( k \) and subset of variables \( a \), a weighted feature \( f_{ak}(z_a) \) with \( \theta_{ak} \).
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:

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

- Difference: not only subsets $\alpha$, but also directions, given by $\pi$. 
Starting Slowly: Naive Bayes
Text Classes

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

\[ \text{TEXT} \leftrightarrow \text{CATEGORY} \]

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

\[ P(\text{text, category}) \]

which quantifies how likely the match between

a text text and a category category is

- For instance,

\[ P(\text{‘I am feeling hungry these days’, ‘poetry’}) \approx 0 \]

\[ P(\text{‘Manchester United’s stock rose after their victory’, ‘business’}) \]

\[ \lor \]

\[ P(\text{‘Manchester United’s stock rose after their victory’, ‘sports’}) \]
Hence, given a sequence of words (including punctuation),

\[ w = (w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, \ldots, w_n) \]

assuming we know \( P \), the joint probability between texts and categories,

an easy way to guess the category of \( w \) is by looking at

\[
\text{category-prediction}(w) = \arg\max_C P(C|w_1, w_2, \ldots, w_n)
\]
\[ P(\text{'poetry'}|\text{`I am feeling hungry these days'}) = 0.0037 \]
\[ P(\text{'business'}|\text{`I am feeling hungry these days'}) = 0.005 \]
\[ P(\text{'sports'}|\text{`I am feeling hungry these days'}) = 0.003 \]
\[ P(\text{'food'}|\text{`I am feeling hungry these days'}) = 0.2 \]
\[ P(\text{'economy'}|\text{`I am feeling hungry these days'}) = 0.04 \]
\[ P(\text{'society'}|\text{`I am feeling hungry these days'}) = 0.08 \]
Text classification & probabilistic framework

\[ P(\text{'poetry'}|\text{I am feeling hungry these days}) = 0.0037 \]
\[ P(\text{'business'}|\text{I am feeling hungry these days}) = 0.005 \]
\[ P(\text{'sports'}|\text{I am feeling hungry these days}) = 0.003 \]
\[ \rightarrow P(\text{'food'}|\text{I am feeling hungry these days}) = 0.2 \]
\[ P(\text{'economy'}|\text{I am feeling hungry these days}) = 0.04 \]
\[ P(\text{'society'}|\text{I am feeling hungry these days}) = 0.08 \]
Bayes Rule

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

\[
P(C|w_1, w_2, \cdots, w_n) = \frac{P(C, w_1, w_2, \cdots, w_n)}{P(w_1, w_2, \cdots, w_n)}
\]

- When looking for the category \( C \) that best fits \( w \), we only focus on the numerator.

- Bayes theorem also gives that

\[
P(C, w_1, \cdots, w_n) = P(C)P(w_1, w_2, \cdots, w_n|C)
\]

\[
= P(C)P(w_1|C)P(w_2, w_3, \cdots, w_n|C, w_1)
\]

\[
= P(C)P(w_1|C)P(w_2|C, w_1)P(w_3, w_4, \cdots, w_n|C, w_1, w_2)
\]

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

- Assume we have the beginning of this news title

  \( w_1, \ldots, w_{12} = \text{‘The weather was so bad that the organizers decided to close the’} \)

- If \( C = \text{business} \), then

  \[
P(W_{13} = \text{‘market’}|\text{business}, w_1, \ldots, w_{12})
  \]

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

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

  \[
P(W_{13} = \text{‘game’}|\text{sports}, w_1, \ldots, w_{12})
  \]
The Naive Bayes Assumption

• From a factorization

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

which handles all the \textbf{conditional} structures of text,

• we assume that each word appears \textbf{independently conditionally} to \( C \),

\[ P(w_i|C, w_1, \ldots, w_{i-1}) = P(w_i|C, w_1, \ldots, w_{i-1}) \]
\[ = P(w_i|C) \]

• and thus

\[ P(C, w_1, \ldots, w_n) = P(C) \prod_{i=1}^{n} P(w_i|C) \]
Naive Bayes & Logistic Regression
Binary Case
Recall the **Naive Bayes Assumption** on $p(x, y)$

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

- Bayes classifier can be interpreted as a **directed** graphical model, where
  - $V = \{X = \{1, \ldots, N\}\} \cup \{Y = 1\}$
  - All elements of $X$ have only one parent:
    \[
    \pi(i) = 1.
    \]
Logistic Regression

- Famous technique for classification (with binary variables):

  **Logistic Regression** (or Maximum Entropy Classifier), model $p(y|x)$

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

- by malaxing things a bit, introducing
  - $f_{y',j}(y, x) = \delta_{y'=y} x_j$
  - $f_{y'}(y, x) = \delta_{y'=y}$

- and renumbering all these functions (and the corresponding weights $\theta_{y,j}$ and $\theta_y$) 1 to $K$,

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

  we obtain an **undirected** graphical model.
A Simple Example: Classification

**Naive Bayes Assumption,** \( p(x, y) \)

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

equivalent to a directed graphical model

**Logistic Regression,** \( p(y|x) \)

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

equivalent to an undirected graphical model
Link between Naive Bayes and Logistic Regression

Deriving the conditional distribution $p(y|x)$ of Naive Bayes

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

- 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

\[ p(x_i = x_i|y = y) = p_{i0}^{(1-y)x_i} p_{i1}^{y x_i} (1-p_{i0})^{(1-y)(1-x_i)} (1-p_{i1})^y (1-x_i) \]

and

\[ p(y = y) = p_1^y (1-p_1)^{1-y} \]

• Define

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

Source: Y.Bulatov
Link between Naive Bayes and Logistic Regression

- then

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

- which can be decomposed again as

\[
p(x, y) = \frac{e^{(\theta_0 + \sum_{i=1}^N \theta_i x_i)y}}{1 + e^{\theta_0 + \sum_{i=1}^N \theta_i x_i}} \times \frac{e^{\sum_{i=1}^N \phi_i x_i} \left(1 + e^{\theta_0 + \sum_{i=1}^N \theta_i x_i}\right)}{\prod_{i=1}^N (1 + e^{\phi_i}) + e^{\theta_0} \prod_{i=1}^N (1 + e^{\theta_i + \phi_i})}
\]

\[
= p(y|x) \times p(x)
\]

- 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 $Y = 1, \ldots, T$ of $T$ words, $X = 1, \ldots, T$

Recall the **Hidden Markov Model** on $p(x, y)$

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

- Of course, HMM’s are **directed** graphical model, where
  - $V = \{X = \{1, \ldots, T\}\} \cup \{Y = \{1, \ldots, T\}\}$
  - Each element of $X$ has only one parent:
    $$\pi(i) = i.$$
  - Each element of $\{2, \ldots, T\}$ has one parent:
    $$\pi(i) = i - 1.$$
The **Linear Conditional Random Field** on $p(y|x)$

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

$$p(y|x) = \frac{1}{Z(x)} \prod_{t=1}^{T} \exp \left\{ \sum_{k=1}^{K} \theta_{k} f_{k}(y_{t}, y_{t-1}, x_{t}) \right\},$$

where $Z(x)$ is an instance-specific normalization function

$$Z(x) = \sum_{y} \prod_{t=1}^{T} \exp \left\{ \sum_{k=1}^{K} \theta_{k} f_{k}(y_{t}, y_{t-1}, x_{t}) \right\}.$$

- The Linear-Chain CRF is an **undirected** graphical model
Let us rewrite the HMM density

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

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

\[ \theta_{ij} = \log p(y' = i | y = j) \quad \text{and} \quad \mu_{oi} = \log p(x = o | y = i). \]
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** \((i, j)\),

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

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

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

● Once this is done, we get

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

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 $p(y|x)$ induced by HMM's

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

- this is the linear CRF induced by HMM's...
Differences between HMM and Linear CRF

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

  However, other types of linear-chain CRFs, \textbf{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 \textbf{for instance} on the current observation vector, \textit{e.g.} by defining
    \[
    f_k = \mathbf{1}_{\{y_t=j\}} \mathbf{1}_{\{y_{t-1}=1\}} \mathbf{1}_{\{x_t=o\}}.
    \]
General CRF

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

- Many parameters potentially...
- For linear chain CRF, same weights/functions are used for factors \( \Psi_t(y_t, y_{t-1}, x_t), \forall t \).
- **Solution**: Partition set of subsets of variables \( \mathcal{F} \) into groups \( \mathcal{F} = \mathcal{F}_1, \ldots, \mathcal{F}_P \).
- Each subset \( \mathcal{F}_i \) is a set of subsets of variables which share the same local functions, i.e.
\[
p(y|x) = \frac{1}{Z(x)} \prod_{\mathcal{F}_i \in \mathcal{F}} \prod_{\Psi_a \in \mathcal{F}_i} \Psi_a(y_a, x_a)
\]
where
\[
\Psi_a(y_a, x_a) = \exp \left\{ \sum_{k=1}^{K(i)} \theta_{ik} f_{ik}(y_a, x_a) \right\}.
\]
- 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,

\[ f_{pk}(y_c, x_c) = 1_{\{y_c = \tilde{y}_c\}} q_{pk}(x_c). \]

- Each feature is **factorized**
  - is nonzero only for a single output configuration \( \tilde{y}_c \),
  - its value only depends input observation \( x_c \).
- 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 \( q_{pk}(x_c) \) are called **observation functions**.
- Examples of observation functions are
  - “word \( x_t \) is capitalized”,
  - “word \( x_t \) ends in \( \text{ing} \)”. 

PRA - 2013

33
Learning with Linear Chain CRF’s
Estimation and Prediction

A linear-chain CRF is a distribution \( p(y|x) \) that takes the form

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

- Two major tasks ahead:
  - Given a set of features \( f_k \), estimate all parameters \( \theta_k \)
  - Predict the labels of a new input \( x \), \( y^* = \arg \max_y p(y|x) \).

- 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

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

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

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

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

\[ p(x) = \sum_y p(x, y) = \sum_y \prod_{t=1}^T \Psi_t(y_t, y_{t-1}, x_t) \]

\[ = \sum_{y_T} \sum_{y_{T-1}} \Psi_T(y_T, y_{T-1}, x_T) \sum_{y_{T-2}} \Psi_{T-1}(y_{T-1}, y_{T-2}, x_{T-1}) \sum_{y_{T-3}} \cdots \]

- Idea: cache intermediate sum which are reused many times during the computation of the outer sum.
• In that sense, define **forward variables** $\alpha_t \in \mathbb{R}^M$ (where $M$ is the number of states),

$$
\alpha_t(j) \overset{\text{def}}{=} p(x_{1...t}, y_t = j)
$$

$$
= \sum_{y_{1...t-1}} \Psi_t(j, y_{t-1}, x_t) \prod_{t'=1}^{t-1} \Psi_{t'}(y_{t'}, y_{t'-1}, x_{t'})
$$

• The summation over $y_{1...t-1}$ ranges over all assignments to $y_1, y_2, \ldots, y_{t-1}$.

• The $\alpha_t$ 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 $p(x) = \sum_{y_T} \alpha_T(y_T)$. 

PRA - 2013
Prediction - Backward

- Define a **backward recursion**, with reverse order: introduce $\beta_t$’s

\[
\beta_t(i) \overset{\text{def}}{=} p(x_{t+1\ldots T} | y_t = i) = \sum_{y_{t+1\ldots T}} \prod_{t'=t+1}^T \Psi_{t'}(y_{t'}, y_{t'-1}, x_{t'}),
\]

and the recursion

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

- **Initialization:** $\beta_T(i) = 1$.

- Analogously to the forward case, $p(x)$ can be computed using the backward variables as

\[
p(x) = \beta_0(y_0) \overset{\text{def}}{=} \sum_{y_1} \Psi_1(y_1, y_0, x_1) \beta_1(y_1).
\]
Prediction - Forward Backward

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

\[ p(y_{t-1}, y_t | x) \]

- Two perspectives can be applied, with identical result:

- Taking first a probabilistic viewpoint we can write

\[
p(y_{t-1}, y_t | x) = \frac{p(x | y_{t-1}, y_t) p(y_t, y_{t-1})}{p(x)}
\]

\[
= \frac{p(x_{(1...t-1)}, y_{t-1}) p(y_t | y_{t-1}) p(x_t | y_t) p(x_{(t+1...T)} | y_t)}{p(x)}
\]

\[
\propto \alpha_{t-1}(y_{t-1}) \Psi_t(y_t, y_{t-1}, x_t) \beta_t(y_t),
\]

where in the second line we have used the fact that \( x_{(1...t-1)} \) is independent from \( x_{(t+1...T)} \) and from \( x_t \) given \( y_{t-1}, y_t \).
Taking a factorization perspective, we see that

\[
p(y_{t-1}, y_t, x) = \Psi_t(y_t, y_{t-1}, x_t)
\]

\[
\left( \sum_{y_{1 \ldots t-2}} \prod_{t' = 1}^{t-1} \Psi_{t'}(y_{t'}, y_{t'-1}, x_{t'}) \right)
\]

\[
\left( \sum_{y_{(t+1) \ldots T}} \prod_{t' = t+1}^{T} \Psi_{t'}(y_{t'}, y_{t'-1}, x_{t'}) \right)
\]

which can be computed from the forward and backward recursions as

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

With \( p(y_{t-1}, y_t, x) \), renormalize over \( y_t, y_{t-1} \) to obtain the desired marginal \( p(y_{t-1}, y_t | x) \).
• To compute the **globally most probable assignment** $y^* = \arg\max_y p(y|x)$,

• we observe that the trick earlier still works if all summations are replaced by maximization.

• This yields the Viterbi recursion:

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

\[
p(y|x) = \frac{1}{Z(x)} \prod_{t=1}^{T} \Psi_t(y_t, y_{t-1}, x_t),
\]

where we define

\[
\Psi_t(y_t, y_{t-1}, x_t) = \exp \left\{ \sum_k \theta_k f_k(y_t, y_{t-1}, x_t) \right\}.
\]

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

- Suppose we have i.i.d training data
  \[ \mathcal{D} = \{x^{(i)}, y^{(i)}\}_{i=1}^{N}, \]

  - each \( x^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \ldots x_T^{(i)}\} \) is a sequence of inputs,
  - each \( y^{(i)} = \{y_1^{(i)}, y_2^{(i)}, \ldots y_T^{(i)}\} \) is a sequence of the desired predictions.

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

\[
\ell(\theta) = \frac{1}{N} \sum_{i=1}^{N} \log p(y^{(i)}|x^{(i)}).
\]

namely,

\[
\ell(\theta) = \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{k=1}^{K} \theta_k f_k(y_t^{(i)}, y_{t-1}^{(i)}, x_t^{(i)}) - \sum_{i=1}^{N} \log Z(x^{(i)}).
\]