Sistemas Inteligentes

Markov Random Fields (Stanford handouts)

José Eduardo Ochoa Luna
Dr. Ciencias - Universidade de São Paulo

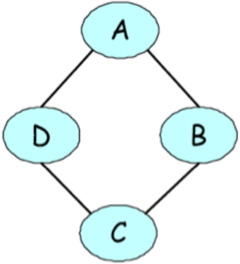
Maestría C.C. Universidad Católica San Pablo
Sistemas Inteligentes

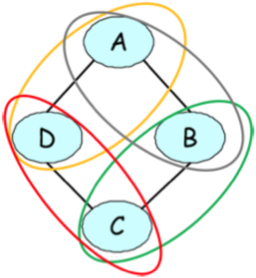
6 de diciembre 2018

Markov Random Fields

Suppose that we are modeling voting preferences among persons A, B, C, D

Let's say that $(A, B), (B, C), (C, D)$ and (D, A) are friends, and friends tend to have similar voting preferences. These influences can be naturally represented by an undirected graph





Markov Random Fields

One way to define a probability over the joint voting decision of A, B, C, D is to assign scores to each assignment to these variables and then define a probability as a normalized score.
A score can be any function, but we will define to be of the form

$$p(A, B, C, D) = \phi(A, B)\phi(B, C)\phi(C, D)\phi(D, A)$$

where $\phi(X, Y)$ is a factor that assigns more weight to consistent votes among friends X, Y

Markov Random Fields

The final probability is defined as

$$p(A, B, C, D) = \frac{1}{Z}\hat{p}(A, B, C, D)$$

where $Z = \sum_{A,B,C,D} \hat{p}(A, B, C, D)$ is a normalizing constant that ensures that the distribution sums to one

Markov Random Fields

- Unlike the directed case, we are not saying anything about how one variable is generated from another set of variables

Markov Random Fields

- Unlike the directed case, we are not saying anything about how one variable is generated from another set of variables
- We simply indicate a level of coupling between dependent variables in the graph

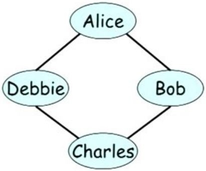
Markov Random Fields

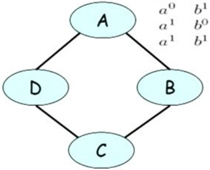
- Unlike the directed case, we are not saying anything about how one variable is generated from another set of variables
- We simply indicate a level of coupling between dependent variables in the graph
- This requires less prior knowledge, as we no longer have to specify a full generative story of how the vote of B is constructed from the vote of A

Markov Random Fields

- Unlike the directed case, we are not saying anything about how one variable is generated from another set of variables
- We simply indicate a level of coupling between dependent variables in the graph
- This requires less prior knowledge, as we no longer have to specify a full generative story of how the vote of B is constructed from the vote of A
- Instead, we simply identify dependent variables and define the strength of their interactions

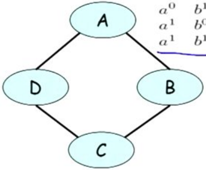
Example: details





$\phi_1[A, B]$

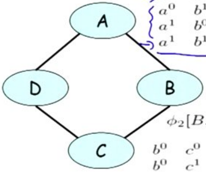
a^0	b^0	30
a^0	b^1	5
a^1	b^0	1
a^1	b^1	10



$\phi_1[A, B]$

a^0	b^0	30
a^0	b^1	5
a^1	b^0	1
a^1	b^1	10

affinity
compatibility
soft constraints



$\phi_1[A, B]$

a^0	b^0	30
a^0	b^1	5
a^1	b^0	1
a^1	b^1	10

affinity
compatibility
soft constraints

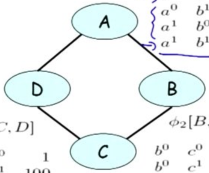
$\phi_2[B, C]$

b^0	c^0	100
b^0	c^1	1
b^1	c^0	1
b^1	c^1	100

$\phi_1[A, B]$

a^0	b^0	30
a^0	b^1	5
a^1	b^0	1
a^1	b^1	10

affinity
compatibility
soft constraints



$\phi_2[B, C]$

b^0	c^0	100
b^0	c^1	1
b^1	c^0	1
b^1	c^1	100

$\phi_3[C, D]$

c^0	d^0	1
c^0	d^1	100
c^1	d^0	100
c^1	d^1	1

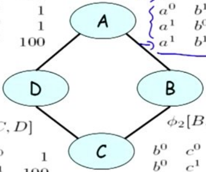
$\phi_4[D, A]$

d^0	a^0	100
d^0	a^1	1
d^1	a^0	1
d^1	a^1	100

$\phi_1[A, B]$

a^0	b^0	30
a^0	b^1	5
a^1	b^0	1
a^1	b^1	10

affinity
compatibility
soft constraints



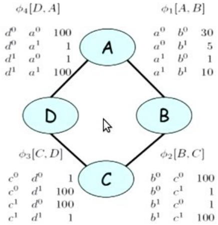
$\phi_3[C, D]$

c^0	d^0	1
c^0	d^1	100
c^1	d^0	100
c^1	d^1	1

$\phi_2[B, C]$

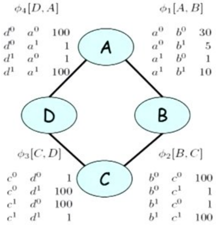
b^0	c^0	100
b^0	c^1	1
b^1	c^0	1
b^1	c^1	100

$$\tilde{P}(A,B,C,D) = \phi_1(A,B) \times \phi_2(B,C) \times \phi_3(C,D) \times \phi_4(A,D)$$



$$\tilde{P}(A,B,C,D) = \phi_1(A,B) \times \phi_2(B,C) \times \phi_3(C,D) \times \phi_4(A,D)$$

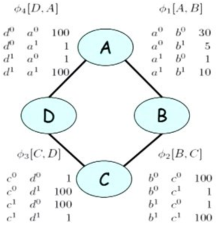
unnormalized measure



$$\tilde{P}(A,B,C,D) = \phi_1(A,B) \times \phi_2(B,C) \times \phi_3(C,D) \times \phi_4(A,D)$$

Unnormalized measure

Assignment				Unnormalized
a^0	b^0	c^0	d^0	300000
a^0	b^0	c^0	d^1	300000
a^0	b^0	c^1	d^0	300000
a^0	b^0	c^1	d^1	30
a^0	b^1	c^0	d^0	500
a^0	b^1	c^0	d^1	500
a^0	b^1	c^1	d^0	5000000
a^0	b^1	c^1	d^1	500
a^1	b^0	c^0	d^0	100
a^1	b^0	c^0	d^1	1000000
a^1	b^0	c^1	d^0	100
a^1	b^0	c^1	d^1	100
a^1	b^1	c^0	d^0	10
a^1	b^1	c^0	d^1	100000
a^1	b^1	c^1	d^0	100000
a^1	b^1	c^1	d^1	100000



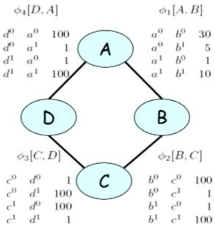
$$\tilde{P}(A,B,C,D) = \phi_1(A,B) \times \phi_2(B,C) \times \phi_3(C,D) \times \phi_4(A,D)$$

unnormalized measure

$$P(A,B,C,D) = \frac{1}{Z}\tilde{P}(A,B,C,D)$$

partition function

Assignment				*Unnormalized*
a^0	b^0	c^0	d^0	300000
a^0	b^0	c^0	d^1	300000
a^0	b^0	c^1	d^0	300000
a^0	b^0	c^1	d^1	30
a^0	b^1	c^0	d^0	500
a^0	b^1	c^0	d^1	500
a^0	b^1	c^1	d^0	5000000
a^0	b^1	c^1	d^1	500
a^1	b^0	c^0	d^0	100
a^1	b^0	c^0	d^1	1000000
a^1	b^0	c^1	d^0	100
a^1	b^0	c^1	d^1	100
a^1	b^1	c^0	d^0	10
a^1	b^1	c^0	d^1	100000
a^1	b^1	c^1	d^0	100000
a^1	b^1	c^1	d^1	100000



$$\tilde{P}(A,B,C,D) = \phi_1(A,B) \times \phi_2(B,C) \times \phi_3(C,D) \times \phi_4(A,D)$$

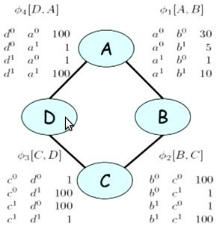
$$P(A,B,C,D) = \frac{1}{Z}\tilde{P}(A,B,C,D)$$

unnormalized measure

partition

Assignment				Unnormalized	Normalized
a^0	b^0	c^0	d^0	300000	0.04
a^0	b^0	c^0	d^1	300000	0.04
a^0	b^0	c^1	d^0	300000	0.04
a^0	b^0	c^1	d^1	30	$4.1 \cdot 10^{-6}$
a^0	b^1	c^0	d^0	500	$6.9 \cdot 10^{-5}$
a^0	b^1	c^0	d^1	500	$6.9 \cdot 10^{-5}$
a^0	b^1	c^1	d^0	5000000	0.69
a^0	b^1	c^1	d^1	500	$6.9 \cdot 10^{-5}$
a^1	b^0	c^0	d^0	100	$1.4 \cdot 10^{-5}$
a^1	b^0	c^0	d^1	1000000	0.14
a^1	b^0	c^1	d^0	100	$1.4 \cdot 10^{-5}$
a^1	b^0	c^1	d^1	100	$1.4 \cdot 10^{-5}$
a^1	b^1	c^0	d^0	10	$1.4 \cdot 10^{-6}$
a^1	b^1	c^0	d^1	100000	0.014
a^1	b^1	c^1	d^0	100000	0.014
a^1	b^1	c^1	d^1	100000	0.014

Z



Daphne Ko

Formal Definition

A Markov Random Field (MRF) is a probability distribution p over variables $x_1, \ldots, x_n$ defined by an undirected graph G in which nodes correspond to variables x_i. The probability p has the form

$$p(x_1, \ldots, x_n) = \frac{1}{Z} \prod_{c \in C} \phi_c(x_c)$$

Where C denotes the set of cliques (i.e. fully connected subgraphs) of G. The value

$$Z = \sum_{x_1, \ldots, x_n} \prod_{c \in C} \phi_c(x_c)$$

is a normalizing constant that ensures that the distribution sums to one.

Formal Definition

- Given a graph G, our probability distribution may contain factors whose scope is any clique in G, which can be a single node, an edge, a triangle, etc.

Formal Definition

- Given a graph G, our probability distribution may contain factors whose scope is any clique in G, which can be a single node, an edge, a triangle, etc.
- Note that we do not need to specify a factor for each clique

Formal Definition

- Given a graph G, our probability distribution may contain factors whose scope is any clique in G, which can be a single node, an edge, a triangle, etc.
- Note that we do not need to specify a factor for each clique
- In our previous example, we defined a factor over each edge (which is a clique of two nodes)

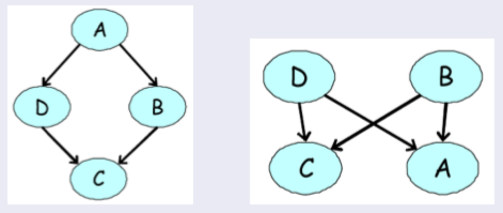
Formal Definition

- Given a graph G, our probability distribution may contain factors whose scope is any clique in G, which can be a single node, an edge, a triangle, etc.
- Note that we do not need to specify a factor for each clique
- In our previous example, we defined a factor over each edge (which is a clique of two nodes)
- However, we chose not to specify any unary factors i.e. cliques over single nodes.

Comparison to Bayesian Networks

In our earlier voting example, we had a distribution over A, B, C, D that satisfied $A \perp C | \{B, D\}$ and $B \perp D | \{A, C\}$ (because only friends directly influence a person's vote).

These independencies cannot be perfectly represented by a Bayesian network.



Comparison to Bayesian Networks

MRFs have several advantages over directed models:

- They can be applied to a wider range of problems in which there is no natural directionality associated with variable dependencies.

Comparison to Bayesian Networks

MRFs have several advantages over directed models:

- They can be applied to a wider range of problems in which there is no natural directionality associated with variable dependencies.
- Undirected graphs can succinctly express certain dependencies that Bayesian nets cannot easily describe (although the converse is also true)

Comparison to Bayesian Networks

They also possess several important drawbacks:

- Computing the normalization constant Z requires summing over a potentially exponential number of assignments.

Comparison to Bayesian Networks

They also possess several important drawbacks:

- Computing the normalization constant Z requires summing over a potentially exponential number of assignments.
- In the general case, this will be NP-hard

Comparison to Bayesian Networks

They also possess several important drawbacks:

- Computing the normalization constant Z requires summing over a potentially exponential number of assignments.
- In the general case, this will be NP-hard
- Many undirected models will be intractable and will require approximation techniques.

Comparison to Bayesian Networks

They also possess several important drawbacks:

- Computing the normalization constant Z requires summing over a potentially exponential number of assignments.
- In the general case, this will be NP-hard
- Many undirected models will be intractable and will require approximation techniques.
- Undirected models may be difficult to interpret.

Comparison to Bayesian Networks

They also possess several important drawbacks:

- Computing the normalization constant Z requires summing over a potentially exponential number of assignments.
- In the general case, this will be NP-hard
- Many undirected models will be intractable and will require approximation techniques.
- Undirected models may be difficult to interpret.
- It is much easier to generate data from a Bayesian network, which is important in some applications.

Comparison to Bayesian Networks

It is not hard to see that Bayesian networks are a special case of MRFs with a very specific type of clique factor and a normalizing constant of one.

In particular, if we take a directed graph G and add side edges to all parents of a given node (and removing their directionality), then the CPDs factorize over the resulting undirected graph. The resulting process is called moralization.

Comparison to Bayesian Networks

- MRFs have more power than Bayesian networks, but are more difficult to deal with computationally.

Comparison to Bayesian Networks

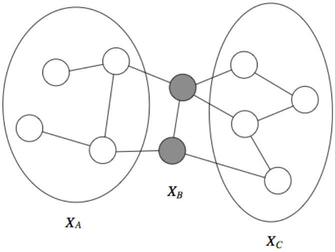
- MRFs have more power than Bayesian networks, but are more difficult to deal with computationally.
- A general rule of thumb is to use Bayesian networks whenever possible, and only switch to MRFs if there is no natural way to model the problem with a directed graph (like in the voting example).

Independencies in Markov Random Fields

Variables x, y are dependent if they are connected by a path of unobserved variables

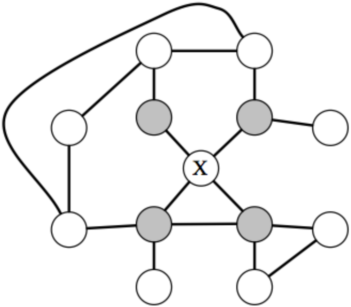
However, if x's neighbors are all observed, then x is independent of all the other variables, since they influence x only via its neighbors.

In particular, if a set of observed variables forms a cutset between two halves of the graph, then variables in one half are independent from ones in the other



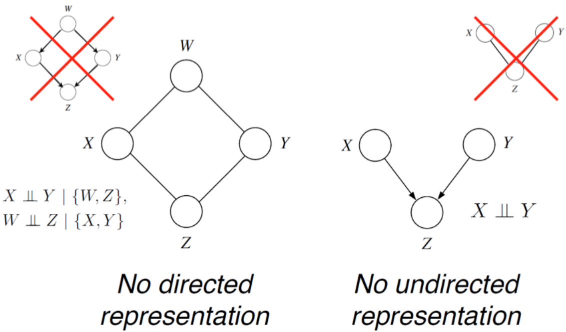
Markov Blanket

Formally, we define the Markov blanket U of a variable X as the minimal set of nodes such that X is independent from the rest of the graph if U is observed, i.e. $X \perp (\mathcal{X} - \{X\} - U) | U$.
This notion holds for both directed and undirected models, but in the undirected case the Markov blanket turns out to simply equal a node's neighborhood.



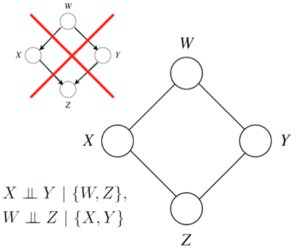
Independencies

- In the directed case, we found that $I(G) \subseteq I(p)$, but there were distributions p whose independencies could not be described by G. In the undirected case, the same holds.

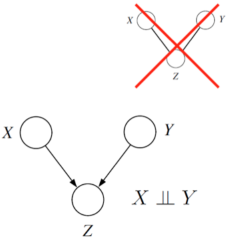


Independencies

- In the directed case, we found that $I(G) \subseteq I(p)$, but there were distributions p whose independencies could not be described by G. In the undirected case, the same holds.
- For example, consider a probability described by a directed v-structure The undirected model cannot describe the independence assumption $X \perp Y$.



No directed representation



No undirected representation

Conditional Random Fields

Conditional Random Fields

- An important special case of MRFs arises when they are applied to model a conditional probability distribution $p(y|x)$

Conditional Random Fields

- An important special case of MRFs arises when they are applied to model a conditional probability distribution $p(y|x)$
- $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ are vector-valued variables

Conditional Random Fields

- An important special case of MRFs arises when they are applied to model a conditional probability distribution $p(y|x)$
- $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ are vector-valued variables
- we are typically given x and want to say something interesting for y.

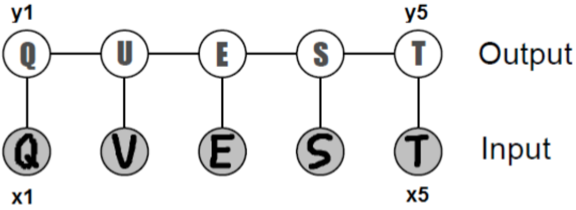
Conditional Random Fields

- An important special case of MRFs arises when they are applied to model a conditional probability distribution $p(y|x)$
- $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ are vector-valued variables
- we are typically given x and want to say something interesting for y.
- Typically, distributions of this sort will arise in a supervised learning setting, where y will be a vector-valued label that we will be trying to predict.

Example

Consider the problem of recognizing a word from a sequence of character images $x_i \in [0,1]^{d \times d}$ given to us in the form of pixel matrices.

The output of our predictor will be a sequence of alphabet letters $y_i \in \{'a', 'b', \ldots, 'z'\}$



Example

- We could in principle train a classifier to separately predict each y_i from its x_i

Example

- We could in principle train a classifier to separately predict each y_i from its x_i
- However, since the letters together form a word, the predictions across different i ought to inform each other

Example

- We could in principle train a classifier to separately predict each y_i from its x_i
- However, since the letters together form a word, the predictions across different i ought to inform each other
- In the example, the second letter by itself could be either a 'U' or a 'V'. Since we can tell with high confidence that its neighbors are 'Q' and 'E', we can infer that 'U' is the most likely true label

Example

- We could in principle train a classifier to separately predict each y_i from its x_i
- However, since the letters together form a word, the predictions across different i ought to inform each other
- In the example, the second letter by itself could be either a 'U' or a 'V'. Since we can tell with high confidence that its neighbors are 'Q' and 'E', we can infer that 'U' is the most likely true label
- CRFs will be a tool that will enable us to perform this prediction jointly.

Formal Definition

Formally, a CRF is Markov network over variables $\mathcal{X} \cup \mathcal{Y}$ which specifies a conditional distribution

$$P(y|x) = \frac{1}{Z(x)} \prod_{c \in C} \phi_c(x_c, y_c)$$

with partition function

$$Z(x) = \sum_{y \in \mathcal{Y}} \prod_{c \in C} \phi_c(x_c, y_c)$$

Formal Definition

- The partition constant now depends on x

Formal Definition

- The partition constant now depends on x
- Which is not surprising: $p(y|x)$ is a probability over y that is parametrized by x,

Formal Definition

- The partition constant now depends on x
- Which is not surprising: $p(y|x)$ is a probability over y that is parametrized by x,
- It encodes a different probability function for each x.

Example

- Suppose $p(y|x)$ is a chain CRF with two types of factors:

Example

- Suppose $p(y|x)$ is a chain CRF with two types of factors:
- Image factors $\phi(x_i, y_i)$ for $i = 1, \ldots, n$, which assign higher values to y_i that are consistent with an input x_i

Example

- Suppose $p(y|x)$ is a chain CRF with two types of factors:
- Image factors $\phi(x_i, y_i)$ for $i = 1, \ldots, n$, which assign higher values to y_i that are consistent with an input x_i
- Pairwise factors $\phi(y_i, y_{i+1})$, for $i = 1, \ldots, n-1$

Example

- Suppose $p(y|x)$ is a chain CRF with two types of factors:
- Image factors $\phi(x_i, y_i)$ for $i = 1, \ldots, n$, which assign higher values to y_i that are consistent with an input x_i
- Pairwise factors $\phi(y_i, y_{i+1})$, for $i = 1, \ldots, n-1$
- We may also think of the $\phi(x_i, y_i)$ as probabilities $p(y_i|x_i)$ given by unstructured softmax regression

Example

- Suppose $p(y|x)$ is a chain CRF with two types of factors:
- Image factors $\phi(x_i, y_i)$ for $i = 1, \dots, n$, which assign higher values to y_i that are consistent with an input x_i
- Pairwise factors $\phi(y_i, y_{i+1})$, for $i = 1, \dots, n-1$
- We may also think of the $\phi(x_i, y_i)$ as probabilities $p(y_i|x_i)$ given by unstructured softmax regression
- The $\phi(y_i, y_{i+1})$ can be seen as empirical frequencies of letter co-occurrences obtained from a large corpus of English text (e.g Wikipedia)

Example

Given a model of this form, we can jointly infer the structured label y using MAP inference:

$$\arg\max_y \phi_1(y_1, x_1) \prod_{i=2}^{n} \phi(y_{i-1}, y_i)\phi(y_i, x_i)$$

CRF Features

In most practical applications, we further assume that the factors $\phi_c(x_c, y_c)$ are of the form

$$\phi_c(x_c, y_c) = \exp(w_c^T f_c(x_c, y_c))$$

where $f_c(x_c, y_c)$ can be an arbitrary set of features describing the compatibility between x_c and y_c

CRF Features

- In the OCR example, we may introduce features $f(x_i, y_i)$ that encode the compatibility of the letter y_i with the pixels x_i

CRF Features

- In the OCR example, we may introduce features $f(x_i, y_i)$ that encode the compatibility of the letter y_i with the pixels x_i
- $f(x_i, y_i)$ could be the probability of letter y_i produced by logistic regression (or a DNN) evaluated on pixels x_i

CRF Features

- In the OCR example, we may introduce features $f(x_i, y_i)$ that encode the compatibility of the letter y_i with the pixels x_i
- $f(x_i, y_i)$ could be the probability of letter y_i produced by logistic regression (or a DNN) evaluated on pixels x_i
- In addition, we introduce features $f(y_i, y_{i+1})$ between adjacent letters

CRF Features

- In the OCR example, we may introduce features $f(x_i, y_i)$ that encode the compatibility of the letter y_i with the pixels x_i
- $f(x_i, y_i)$ could be the probability of letter y_i produced by logistic regression (or a DNN) evaluated on pixels x_i
- In addition, we introduce features $f(y_i, y_{i+1})$ between adjacent letters
- These may be indicators of the form $f(y_i, y_{i+1}) = I(y_i = \mathcal{C}_1, y_{i+1} = \mathcal{C}_2)$, where $\mathcal{C}_1, \mathcal{C}_2$ are two letter of the alphabet

CRF Features

- The CRF would then learn weights w that would assign more weight to more common probability of consecutive letters $\mathcal{C}_1, \mathcal{C}_2,$

CRF Features

- The CRF would then learn weights w that would assign more weight to more common probability of consecutive letters $\mathcal{C}_1, \mathcal{C}_2,$
- while at the same time making sure that the predicted y_i are consistent with the input x_i

CRF Features

- The CRF would then learn weights w that would assign more weight to more common probability of consecutive letters $\mathcal{C}_1, \mathcal{C}_2,$
- while at the same time making sure that the predicted y_i are consistent with the input x_i
- This process would let us determine y_i in cases where x_i is ambiguous

CRF Features

- CRF features can be arbitrarily complex

CRF Features

- CRF features can be arbitrarily complex
- We may define an OCR model with factors $\phi_i(x, y_i) = \exp(w_i^T f(x, y_i))$, that depend on the entire input x

CRF Features

- CRF features can be arbitrarily complex
- We may define an OCR model with factors $\phi_i(x, y_i) = \exp(w_i^T f(x, y_i))$, that depend on the entire input x
- This will not affect computational performance, because at inference time, the x will be always observed and the problem will involve maximizing

$$\phi_1(y_1, x) \prod_{i=2}^{n} \phi(y_{i-1}, y_i)\phi(y_i, x) = \phi_1'(y_1) \prod_{i=2}^{n} \phi(y_{i-1}, y_i)\phi'(y_i)$$

CRF Features

- CRF features can be arbitrarily complex
- We may define an OCR model with factors $\phi_i(x, y_i) = \exp(w_i^T f(x, y_i))$, that depend on the entire input x
- This will not affect computational performance, because at inference time, the x will be always observed and the problem will involve maximizing

$$\phi_1(y_1, x) \prod_{i=2}^{n} \phi(y_{i-1}, y_i)\phi(y_i, x) = \phi'_1(y_1) \prod_{i=2}^{n} \phi(y_{i-1}, y_i)\phi'(y_i)$$

- where $\phi'_i(y_i) = \phi_i(x, y_i)$.

CRF Features

- If we were to model $p(x, y)$ using an MRF, then we need to fit two distributions to the data $p(y|x)$ and $p(x)$

CRF Features

- If we were to model $p(x, y)$ using an MRF, then we need to fit two distributions to the data $p(y|x)$ and $p(x)$
- If we are interested in predicting y given x, then modeling $p(x)$ is unnecessary

CRF Features

- If we were to model $p(x, y)$ using an MRF, then we need to fit two distributions to the data $p(y|x)$ and $p(x)$
- If we are interested in predicting y given x, then modeling $p(x)$ is unnecessary
- In fact, it may be disadvantageous to do so statistically and it may not be a good idea computationally

CRF Features

- If we were to model $p(x, y)$ using an MRF, then we need to fit two distributions to the data $p(y|x)$ and $p(x)$
- If we are interested in predicting y given x, then modeling $p(x)$ is unnecessary
- In fact, it may be disadvantageous to do so statistically and it may not be a good idea computationally
- CRFs forgot of this assumption and often perform better on prediction tasks