PGM 3 — Learning parameters

Qinfeng (Javen) Shi

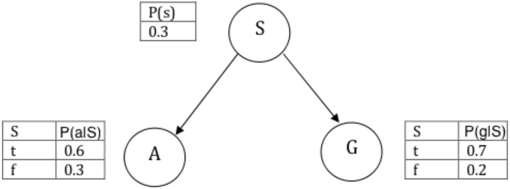
ML session 11

Table of Contents I

1. Learning parameters for Bayes Net
   - Example
   - Parameters for Bayes Net?
   - Learn from the data (Bayes Net, Naive Bayes)?

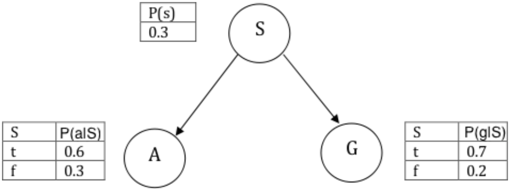
2. Learning parameters for MRFs
   - Parameters for MRFs
   - Max Margin Approaches (e.g. Structured SVMs)
   - Probabilistic Approaches (e.g. CRFs)

Example of 4WD



- $P(A)$? (marginal inference)
- $\mathrm{argmax}_{G,A,S}\, P(G, A, S)$? (MAP inference)

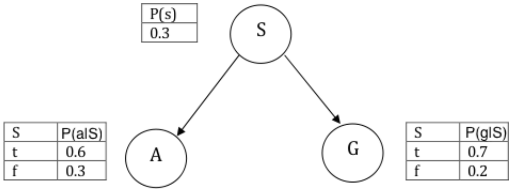
Example of 4WD



- $P(A)$? (marginal inference)
- $\operatorname{argmax}_{G,A,S} P(G, A, S)$? (MAP inference)

How to get the local probability tables (called parameters)?

Example of 4WD



P(s)
0.3

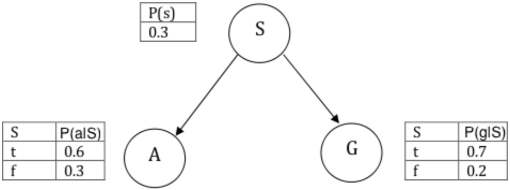
S	P(a|S)
t	0.6
f	0.3

S	P(g|S)
t	0.7
f	0.2

- $P(A)$? (marginal inference)
- $\mathrm{argmax}_{G,A,S}\, P(G, A, S)$? (MAP inference)

How to get the local probability tables (called parameters)?
$\rightarrow$ Learning parameters (today)

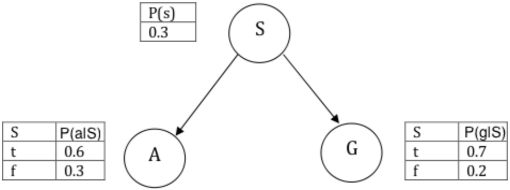
Example of 4WD



- $P(A)$? (marginal inference)
- $\operatorname{argmax}_{G,A,S} P(G, A, S)$? (MAP inference)

How to get the local probability tables (called parameters)?
$\rightarrow$ Learning parameters (today)
How to get the graph?

Example of 4WD



- $P(A)$? (marginal inference)
- $\mathrm{argmax}_{G,A,S}\, P(G, A, S)$? (MAP inference)

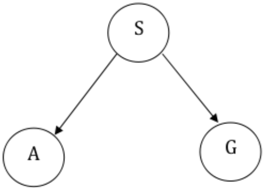
How to get the local probability tables (called parameters)?
$\rightarrow$ Learning parameters (today)
How to get the graph?
$\rightarrow$ Learning structures (today)

What are the parameters for Bayes Net?

For bayesian networks, $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | Pa(x_i))$.
Parameters: $P(x_i | Pa(x_i))$.

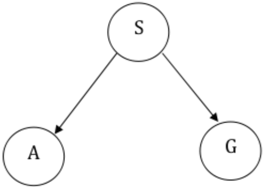
What are the parameters for Bayes Net?

For bayesian networks, $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | Pa(x_i))$.
Parameters: $P(x_i | Pa(x_i))$.



What are the parameters for Bayes Net?

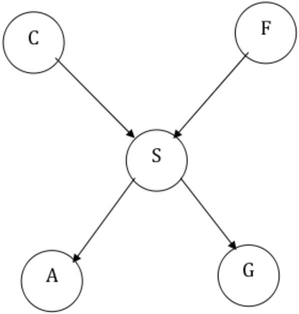
For bayesian networks, $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | Pa(x_i))$.
Parameters: $P(x_i | Pa(x_i))$.



Parameters: $P(S), P(A|S), P(G|S)$

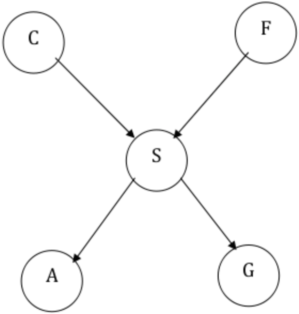
What are the parameters for Bayes Net?

Parameters?



What are the parameters for Bayes Net?

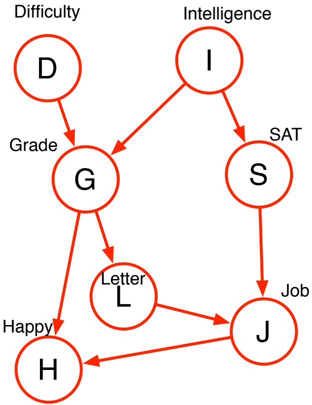
Parameters?



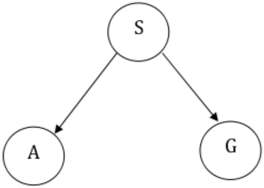
Parameters: $P(C), P(F), P(S|C, F), P(A|S), P(G|S)$

What are the parameters for Bayes Net?

Parameters?



How to learn the parameters from the data?



Parameters: $P(S), P(A|S), P(G|S)$

Data:

S	A	G
1	0	0
0	0	1
1	1	0
1	1	0

How to learn the parameters from the data?

Parameters: $P(S), P(A|S), P(G|S)$
Data:

S	A	G
1	0	0
0	0	1
1	1	0
1	1	0

How to learn the parameters from the data?

Parameters: $P(S), P(A|S), P(G|S)$

Data:

S	A	G
1	0	0
0	0	1
1	1	0
1	1	0

$$P(S=0) \approx \frac{N_{(S=0)}}{N_{(S=0)} + N_{(S=1)}} = \frac{1}{4}$$

$$P(S=1) \approx \frac{N_{(S=1)}}{N_{(S=0)} + N_{(S=1)}} = \frac{3}{4}$$

$$P(A=0|S=0) \approx \frac{N_{(A=0,S=0)}}{N_{(S=0)}} = \frac{1}{1}$$

...

Problem?

Parameters: $P(S), P(A|S), P(G|S)$
What if we change the data only by one entry (instance)?

S	A	G		S	A	G
1	0	0		1	0	0
0	0	1	$\rightarrow$	1	0	1
1	1	0		1	1	0
1	1	0		1	1	0

Problem?

Parameters: $P(S), P(A|S), P(G|S)$
What if we change the data only by one entry (instance)?

S	A	G		S	A	G
1	0	0		1	0	0
0	0	1	$\rightarrow$	1	0	1
1	1	0		1	1	0
1	1	0		1	1	0

$$P(A = 0|S = 0) \approx \frac{N_{(A=0,S=0)}}{N_{(S=0)}} = \frac{0}{0} \quad ?!$$

Problem?

Parameters: $P(S), P(A|S), P(G|S)$

$$P(A = 0|S = 0) \approx \frac{N_{(A=0,S=0)}}{N_{(S=0)}} = \frac{0}{0}$$

Problem?

Parameters: $P(S), P(A|S), P(G|S)$

$$P(A=0|S=0) \approx \frac{N_{(A=0,S=0)}}{N_{(S=0)}} = \frac{0}{0}$$

Solution 1: set $P(A|S=0)$ to be uniform distribution when $N_{(S=0)} = 0$.

Problem?

Parameters: $P(S), P(A|S), P(G|S)$

$$P(A=0|S=0) \approx \frac{N_{(A=0,S=0)}}{N_{(S=0)}} = \frac{0}{0}$$

Solution 1: set $P(A|S=0)$ to be uniform distribution when $N_{(S=0)} = 0$. Why?

Problem?

Parameters: $P(S), P(A|S), P(G|S)$

$$P(A=0|S=0) \approx \frac{N_{(A=0,S=0)}}{N_{(S=0)}} = \frac{0}{0}$$

Solution 1: set $P(A|S=0)$ to be uniform distribution when $N_{(S=0)} = 0$. Why?

Problem?

Parameters: $P(S), P(A|S), P(G|S)$

$$P(A=0|S=0) \approx \frac{N_{(A=0,S=0)}}{N_{(S=0)}} = \frac{0}{0}$$

Solution 1: set $P(A|S=0)$ to be uniform distribution when $N_{(S=0)} = 0$. Why?

Solution 2 (better): set (no need to check if the denominator)

$$P(A=0|S=0) \approx \frac{N_{(A=0,S=0)} + N_r}{N_{(S=0)} + (\#A) \times N_r}$$

Often $N_r = 1$. $\#A$ is the number of values of variable A can take.

General solution when the denominator is 0

Let $A, B, C, D, \ldots$ be the variables. To estimate $P(A = 0|B = 0, C = 0)$.

$$P(A = 0|B = 0, C = 0) \approx \frac{N_{(A=0,B=0,C=0)}}{N_{(B=0,C=0)}}$$

What if $N_{(B=0,C=0)} = 0$? This means $N_{(A=0,B=0,C=0)} = 0$ and $N_{(A=1,B=0,C=0)} = 0$.

Solution 1: When this happens, set $P(A|B = 0, C = 0)$ to be uniform distribution.

Solution 2 (better): Always set (no need to check if the denominator $= 0$ or not)

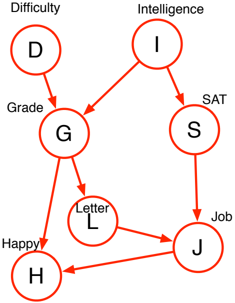
$$P(A = 0|B = 0, C = 0) \approx \frac{N_{(A=0,B=0,C=0)} + N_r}{N_{(B=0,C=0)} + (\#A) \times N_r}$$

Often $N_r = 1$. $\#A$ is the number of values of variable A can take.

A general case (Student model)

Data:

D	I	G	S	L	H	J
1	0	0	1	0	1	0
0	0	1	0	0	0	0
1	1	0	0	0	1	1
⋮						



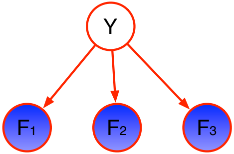
$$P(D=0) = \frac{N_{(D=0)}}{N_{total}}$$

$$P(D=1) = \frac{N_{(D=1)}}{N_{total}}$$

$$P(G=0|D=0, I=1) = \frac{N_{(G=0,D=0,I=1)}}{N_{(D=0,I=1)}}$$

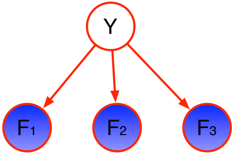
⋮

A special case (Naive Bayes)

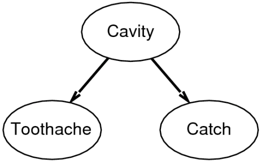


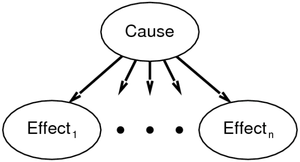
Parameters: $P(F_1|Y), P(F_2|Y), \ldots$

A special case (Naive Bayes)



Parameters: $P(F_1|Y), P(F_2|Y), \ldots$





More problems?

- not minimise classification error or other measure of your task.
- not much flexibility on the features nor the parameters.

More problems?

- not minimise classification error or other measure of your task.
- not much flexibility on the features nor the parameters.

Alternatives: using MRFs or factor graphical models.

Parameters for MRFs

For MRFs, let $\mathcal{V}$ be the set of nodes, and $\mathcal{C}$ be the set of clusters c.

$$P(\mathbf{x};\theta) = \frac{\exp(\sum_{c\in\mathcal{C}} \theta_c(\mathbf{x}_c))}{Z(\theta)}, \tag{1}$$

where normaliser $Z(\theta) = \sum_{\mathbf{x}} \exp\{\sum_{c''\in\mathcal{C}} \theta_{c''}(\mathbf{x}_{c''})\}$.
Parameters: $\{\theta_c\}_{c\in\mathcal{C}}$ or $\mathbf{w}$.

- Often assume $\theta_c(\mathbf{x}_c) = \langle \mathbf{w}, \Phi_c(\mathbf{x}_c)\rangle$.
- $\mathbf{w} \leftarrow$ empirical risk minimisation (ERM).

Inference:

- MAP inference $\mathbf{x}^* = \operatorname{argmax}_{\mathbf{x}} \sum_{c\in\mathcal{C}} \theta_c(\mathbf{x}_c)$
(hint: $\log P(\mathbf{x}) \propto \sum_{c\in\mathcal{C}} \theta_c(\mathbf{x}_c)$)
- Marginal inference $P(\mathbf{x}_c) = \sum_{\mathbf{x}_{\mathcal{V}/c}} P(\mathbf{x})$

Parameters for MRFs

In learning, we look for a F that predicts labels well via

$$\mathbf{y}^* = \max_{\mathbf{y}\in\mathcal{Y}} F(\mathbf{x}_i, \mathbf{y}; \mathbf{w}).$$

Given graph $G = (V, E)$, one often assume

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}, \qquad \Phi(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} \sum_{i\in V} \Phi_i(y^{(i)}, \mathbf{x}) \\ \sum_{(i,j)\in E} \Phi_{i,j}(y^{(i)}, y^{(j)}, \mathbf{x})] \end{bmatrix}$$

$$\begin{aligned} F(\mathbf{x}, \mathbf{y}; \mathbf{w}) &= \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle \\ &= \sum_{i\in V} \left\langle \mathbf{w}_1, \Phi_i(y^{(i)}, \mathbf{x}) \right\rangle + \sum_{(i,j)\in E} \left\langle \mathbf{w}_2, \Phi_{i,j}(y^{(i)}, y^{(j)}, \mathbf{x}) \right\rangle \\ &= \sum_{i\in V} \theta_i(y^{(i)}, \mathbf{x}) + \sum_{(i,j)\in E} \theta_{i,j}(y^{(i)}, y^{(j)}, \mathbf{x}) \end{aligned}$$

Max Margin Approaches

A gap between $F(\mathbf{x}_i, \mathbf{y}_i; \mathbf{w})$ and best $F(\mathbf{x}_i, \mathbf{y}; \mathbf{w})$ for $\mathbf{y} \neq \mathbf{y}_i$, that is

$$F(\mathbf{x}_i, \mathbf{y}_i; \mathbf{w}) - \max_{\mathbf{y} \in \mathcal{Y}, \mathbf{y} \neq \mathbf{y}_i} F(\mathbf{x}_i, \mathbf{y}; \mathbf{w})$$

Structured SVM - 1

Primal:

$$\min_{\mathbf{w},\xi} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i \quad \text{s.t.} \tag{2a}$$

$$\forall i, \mathbf{y} \neq \mathbf{y}_i, \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \mathbf{y})\rangle \geq \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i. \tag{2b}$$

Dual is a quadratic programming (QP) problem similar to binary SVM's dual.

Structured SVM - 2

Cutting plane method needs to find the label for the most violated constraint in (2b)

$$\mathbf{y}_i^\dagger = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}} \Delta(\mathbf{y}_i, \mathbf{y}) + \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}) \rangle . \tag{3}$$

With $\mathbf{y}_i^\dagger$, one can solve following relaxed problem (with much fewer constraints)

$$\min_{\mathbf{w},\xi} \ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{m} \xi_i \quad \text{s.t.} \tag{4a}$$

$$\forall i, \left\langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \mathbf{y}_i^\dagger) \right\rangle \geq \Delta(\mathbf{y}_i, \mathbf{y}_i^\dagger) - \xi_i . \tag{4b}$$

Structured SVM - 3

Input: data $\mathbf{x}_i$, labels $\mathbf{y}_i$, sample size m, number of iterations T
Initialise $S_0 = \emptyset$, $\mathbf{w}_0 = 0$ (or a random vector), and $t = 0$.
for $t = 0$ **to** T **do**
  for $i = 1$ **to** m **do**
    $\mathbf{y}_i^\dagger = \mathrm{argmax}_{\mathbf{y} \in \mathcal{Y}, \mathbf{y} \neq \mathbf{y}_i} \langle \mathbf{w}_t, \Phi(\mathbf{x}_i, \mathbf{y}) \rangle + \Delta(\mathbf{y}_i, \mathbf{y})$,
    $\xi_i = \left[\Delta(\mathbf{y}_i, \mathbf{y}) + \left\langle \mathbf{w}_t, \left(\Phi(\mathbf{x}_i, \mathbf{y}_i^\dagger) - \Phi(\mathbf{x}_i, \mathbf{y}_i) \right) \right\rangle \right]_+$,
    if $\xi_i > 0$ **then**
      Increase constraint set $S_t \leftarrow S_t \cup \{\mathbf{y}_i^\dagger\}$
    end if
  end for
  $\mathbf{w}_t$ recovered using dual variables.
  $\alpha \leftarrow$ optimise dual QP with constraint set S_t.
end for

Other Max Margin Approaches

Other approaches using Max Margin principle such as Max Margin Markov Network (M3N), ...

Probabilistic Approaches

Main types:

- Maximum Entropy (MaxEnt)
- Maximum a Posteriori (MAP)
- Maximum Likelihood (ML)

Maximum Entropy

Maximum Entropy (ME) estimates $\mathbf{w}$ by maximising the entropy. That is,

$$\mathbf{w}^* = \operatorname*{argmax}_{\mathbf{w}} \sum_{\mathbf{x}\in\mathcal{X},\mathbf{y}\in\mathcal{Y}} -\mathbf{P}_{\mathbf{w}}(\mathbf{x},\mathbf{y}) \ln \mathbf{P}_{\mathbf{w}}(\mathbf{x},\mathbf{y}).$$

Duality between maximum likelihood, and maximum entropy, subject to moment matching constraints on the expectations of features.

MAP

Let likelihood function $\mathcal{L}(\mathbf{w})$ be the modelled probability or density for the occurrence of a sample configuration $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_m, \mathbf{y}_m)$ given the probability density $\mathbf{P}_\mathbf{w}$ parameterised by $\mathbf{w}$. That is,

$$\mathcal{L}(\mathbf{w}) = \mathbf{P}_\mathbf{w}\left((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_m, \mathbf{y}_m)\right).$$

Maximum a Posteriori (MAP) estimates $\mathbf{w}$ by maximising $\mathcal{L}(\mathbf{w})$ times a prior $P(\mathbf{w})$. That is

$$\mathbf{w}^* = \operatorname*{argmax}_\mathbf{w} \mathcal{L}(\mathbf{w}) P(\mathbf{w}). \tag{5}$$

Assuming $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{1 \leq i \leq m}$ are I.I.D. samples from $\mathbf{P}_\mathbf{w}(\mathbf{x}, \mathbf{y})$, (5) becomes

$$\begin{aligned} \mathbf{w}^* &= \operatorname*{argmax}_\mathbf{w} \prod_{1 \leq i \leq m} \mathbf{P}_\mathbf{w}(\mathbf{x}_i, \mathbf{y}_i) P(\mathbf{w}) \\ &= \operatorname*{argmin}_\mathbf{w} \sum_{1 \leq i \leq m} - \ln \mathbf{P}_\mathbf{w}(\mathbf{x}_i, \mathbf{y}_i) - \ln P(\mathbf{w}). \end{aligned}$$

Maximum Likelihood

Maximum Likelihood (ML) is a special case of MAP when $P(\mathbf{w})$ is uniform which means

$$\begin{aligned}\mathbf{w}^* &= \operatorname*{argmax}_{\mathbf{w}} \prod_{1 \le i \le m} \mathbf{P}_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}_i) \\ &= \operatorname*{argmin}_{\mathbf{w}} \sum_{1 \le i \le m} -\ln \mathbf{P}_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}_i).\end{aligned}$$

Alternatively, one can replace the joint distribution $\mathbf{P}_{\mathbf{w}}(\mathbf{x}, \mathbf{y})$ by the conditional distribution $\mathbf{P}_{\mathbf{w}}(\mathbf{y} \mid \mathbf{x})$ that gives a discriminative model called Conditional Random Fields (CRFs)

Conditional Random Fields (CRFs) - 1

Assume the conditional distribution over $\mathcal{Y} \,|\, \mathcal{X}$ has a form of exponential families, *i.e.*,

$$\mathbf{P}(\mathbf{y} \,|\, \mathbf{x}; \mathbf{w}) = \frac{\exp(\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle)}{Z(\mathbf{w}, \mathbf{x})}, \tag{6}$$

where

$$Z(\mathbf{w}, \mathbf{x}) = \sum_{\mathbf{y}' \in \mathcal{Y}} \exp(\big\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}') \big\rangle), \tag{7}$$

and

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}, \qquad \Phi(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} \sum_{i \in V} \Phi_i(y^{(i)}, \mathbf{x}) \\ \sum_{(i,j) \in E} \Phi_{i,j}(y^{(i)}, y^{(j)}, \mathbf{x})] \end{bmatrix}$$

More generally speaking, the global feature can be decomposed into local features on cliques (fully connected subgraphs).

CRFs - 2

Denote $(\mathbf{x}_1, \ldots, \mathbf{x}_m)$ as $\mathbf{X}$, $(\mathbf{y}_1, \ldots, \mathbf{y}_m)$ as $\mathbf{Y}$. The classical approach is to maximise the conditional likelihood of $\mathbf{Y}$ on $\mathbf{X}$, incorporating a prior on the parameters. This is a Maximum a Posteriori (MAP) estimator, which consists of maximising

$$\mathbf{P}(\mathbf{w} \,|\, \mathbf{X}, \mathbf{Y}) \propto P(\mathbf{w})\, \mathbf{P}(\mathbf{Y} \,|\, \mathbf{X}; \mathbf{w}).$$

From the i.i.d. assumption we have

$$\mathbf{P}(\mathbf{Y} \,|\, \mathbf{X}; \mathbf{w}) = \prod_{i=1}^{m} \mathbf{P}(\mathbf{y}_i \,|\, \mathbf{x}_i; \mathbf{w}),$$

and we impose a Gaussian prior on $\mathbf{w}$

$$P(\mathbf{w}) \propto \exp\left(\frac{-\|\,\mathbf{w}\,\|^2}{2\sigma^2}\right).$$

CRFs - 3

Maximising the posterior distribution can also be seen as minimising the negative log-posterior, which becomes our risk function $R(\mathbf{w}, \mathbf{X}, \mathbf{Y})$

$$\begin{aligned} R(\mathbf{w}, \mathbf{X}, \mathbf{Y}) &= -\ln(P(\mathbf{w})\,\mathbf{P}(\mathbf{Y} \mid \mathbf{X}; \mathbf{w})) + c \\ &= \frac{\|\mathbf{w}\|^2}{2\sigma^2} - \sum_{i=1}^{m} \underbrace{\Big(\langle \Phi(\mathbf{x}_i, \mathbf{y}_i), \mathbf{w} \rangle - \ln(Z(\mathbf{w}, \mathbf{x}_i)) \Big)}_{:=\ell_L(\mathbf{x}_i, \mathbf{y}_i, \mathbf{w})} + c, \end{aligned}$$

where c is a constant and ℓ_L denotes the log loss *i.e.* negative log-likelihood. Now learning is equivalent to

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} R(\mathbf{w}, \mathbf{X}, \mathbf{Y}).$$

CRFs - 4

Above is a convex optimisation problem on $\mathbf{w}$ since $\ln Z(\mathbf{w}, \mathbf{x})$ is a convex function of $\mathbf{w}$. The solution can be obtained by gradient descent since $\ln Z(\mathbf{w}, \mathbf{x})$ is also differentiable. We have

$$\nabla_{\mathbf{w}} R(\mathbf{w}, \mathbf{X}, \mathbf{Y}) = \frac{\mathbf{w}}{\sigma^2} - \sum_{i=1}^{m} \Big(\Phi(\mathbf{x}_i, \mathbf{y}_i) - \nabla_{\mathbf{w}} \ln(Z(\mathbf{w}, \mathbf{x}_i)) \Big).$$

It follows from direct computation that

$$\nabla_{\mathbf{w}} \ln(Z(\mathbf{w}, \mathbf{x})) = \mathbb{E}_{\mathbf{y} \sim \mathbf{P}(\mathbf{y} \mid \mathbf{x}; \mathbf{w})}[\Phi(\mathbf{x}, \mathbf{y})].$$

CRFs - 5

Since $\Phi(\mathbf{x}, \mathbf{y})$ are decomposed over nodes and edges, it is straightforward to show that the expectation also decomposes into expectations on nodes $\mathcal{V}$ and edges $\mathcal{E}$

$$\mathbb{E}_{\mathbf{y} \sim \mathbf{P}(\mathbf{y} \,|\, \mathbf{x}; \mathbf{w})}[\Phi(\mathbf{x}, \mathbf{y})] =$$
$$\sum_{i \in \mathcal{V}} \mathbb{E}_{y^{(i)} \sim \mathbf{P}(y^{(i)} |\, \mathbf{x}; \mathbf{w})}[\Phi_i(y^{(i)}, \mathbf{x})]$$
$$+ \sum_{(ij) \in \mathcal{E}} \mathbb{E}_{y^{(i)}, y^{(j)} \sim \mathbf{P}(y^{(i)}, y^{(j)} |\, \mathbf{x}; \mathbf{w})}[\Phi_{i,j}(y^{(i)}, y^{(j)}\, \mathbf{x})],$$

where the node and edge expectations can be computed given $\mathbf{P}(y^{(i)} |\, \mathbf{x}; \mathbf{w})$ and $\mathbf{P}(y^{(i)}, y^{(j)} |\, \mathbf{x}; \mathbf{w})$, which can be computed by Marginal inference methods such as variable elimination, junction tree, *e.g.* (loopy) belief propagation, or being circumvented through sampling.