# Lecture 9: PGM — Learning

Qinfeng (Javen) Shi

13 Oct 2014

Intro. to Stats. Machine Learning
COMP SCI 4401/7401

# Table of Contents I

# Inference and Learning

Given parameters (of potentials) and the graph, one can ask for:

- $\mathbf{x}^* = \operatorname{argmax}_{\mathbf{x}} P(\mathbf{x})$ MAP Inference
- $P(\mathbf{x}_c) = \sum_{\mathbf{x}_{V/c}} P(\mathbf{x})$ Marginal Inference

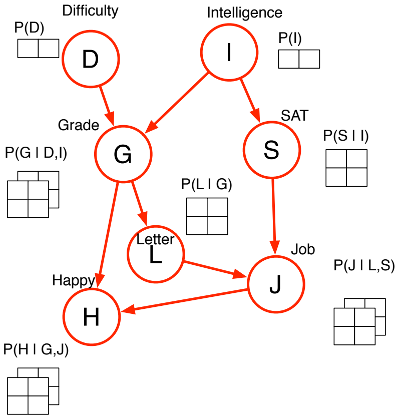
How to get parameters and the graph? $\rightarrow$ Learning.

# Learning

- Learn parameters if graph given (Lecture 9)
  - Bayes Net (Directed graphical models)
  - Markov Random Fields (Undirected or factor graphical models)
- Structure estimation ( to learn or estimate the graph structure, Lecture 10)

# Parameters for bayesian networks

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



# Learning parameters in Bayes Net

Y = Yes. N = No.

| Case | D | I | G | S | L | H | J |
|---|---|---|---|---|---|---|---|
| 1 | Y | Y | Y | Y | Y | N | Y |
| 2 | N | N | Y | N | N | Y | N |
| 3 | Y | N | Y | N | N | Y | N |
| ⋮ | | | | | | | |

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

$$P(G = g | D = d, I = i) = \frac{N_{G=g,D=d,I=i}}{N_{D=d,I=i}}$$

⋮

# Learning parameters in Bayes Net

Problems?

# Learning parameters in Bayes Net

Problems?

- not minimise classification error.
- not much flexibility on the features nor the parameters.

# 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: $\{\theta_c\}_{c\in\mathcal{C}}$.
Inference:

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

Learning (parameter estimation): learn $\theta$ and the graph structure.

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

# 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

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

Here $\mathbf{w} = [\mathbf{w}_1; \mathbf{w}_2]$, and
$\Phi(\mathbf{x}, \mathbf{y}) = [\sum_{i\in V} \Phi_i(y^{(i)}, \mathbf{x}); \sum_{(i,j)\in E} \Phi_{i,j}(y^{(i)}, y^{(j)}, \mathbf{x})]$.

# 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:

$$\max_{\alpha} \; \sum_{i,\mathbf{y}\neq\mathbf{y}_i} \Delta(\mathbf{y}_i, \mathbf{y})\alpha_{i\,\mathbf{y}} - \frac{1}{2}\sum_{i,j,\mathbf{y}\neq\mathbf{y}_i,\mathbf{y}'\neq\mathbf{y}_j} \alpha_{i\,\mathbf{y}}\alpha_{j\,\mathbf{y}'}\left\langle \Phi(\mathbf{x}_i, \mathbf{y}), \Phi(\mathbf{x}_j, \mathbf{y}')\right\rangle$$

$$\forall i, \mathbf{y} \neq \mathbf{y}_i, \;\; \alpha_{i\,\mathbf{y}} \geq 0,$$

$$\forall i, \;\; \sum_{\mathbf{y}\neq\mathbf{y}_i} \alpha_{i\,\mathbf{y}} \leq C. \tag{3}$$

# Structured SVM - 2

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

$$\mathbf{y}_i^{\dagger} = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}} \Delta(\mathbf{y}_i, \mathbf{y}) + \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}) \rangle . \tag{4}$$

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

$$\min_{\mathbf{w}, \xi} \ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{m} \xi_i \quad \text{s.t.} \tag{5a}$$

$$\forall i, \left\langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \mathbf{y}_i^{\dagger}) \right\rangle \geq \Delta(\mathbf{y}_i, \mathbf{y}_i^{\dagger}) - \xi_i . \tag{5b}$$

# Structured SVM - 3

Simplified over all procedure.

**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**
    $\mathbf{y}_i^\dagger = \mathrm{argmax}_{\mathbf{y} \in \mathcal{Y}, \mathbf{y} \neq \mathbf{y}_i} \langle \mathbf{w}_t, \Phi(\mathbf{x}_i, \mathbf{y}) \rangle + \Delta(\mathbf{y}_i, \mathbf{y})$,
    $\xi_i = \left[ \Delta(\mathbf{y}_i, \mathbf{y}) + \left\langle \mathbf{w}_t, \Phi(\mathbf{x}_i, \mathbf{y}_i^\dagger) - \Phi(\mathbf{x}_i, \mathbf{y}_i) \right\rangle \right]_+$,
    **if** $\xi_i > 0$ **then**
      Increase constraint set $S_t \leftarrow S_t \cup \{\mathbf{y}_i^\dagger\}$
    **end if**
  **end for**
  $\mathbf{w}_t = \sum_i \sum_{y \in S_t} \alpha_{i\,\mathbf{y}} \Phi(\mathbf{x}_i, \mathbf{y})$
  $\alpha \leftarrow$ 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

$$\mathbf{w}^* = \operatorname*{argmax}_{\mathbf{w}} \mathcal{L}(\mathbf{w}) P(\mathbf{w}). \tag{6}$$

Assuming $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{1 \le i \le m}$ are I.I.D. samples from $\mathbf{P}_\mathbf{w}(\mathbf{x}, \mathbf{y})$, (6) becomes

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

# Maximum Likelihood

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

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

Alternatively, one can replace the joint distribution $\mathbf{P}_{\mathbf{w}}(\mathbf{x}, \mathbf{y})$ by the conditional distribution $\mathbf{P}_{\mathbf{w}}(\mathbf{y} \,|\, \mathbf{x})$ 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.*,

$$\mathbf{P}(\mathbf{y} \,|\, \mathbf{x}; \mathbf{w}) = \frac{\exp(\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle)}{Z(\mathbf{w} \,|\, \mathbf{x})}, \tag{7}$$

where

$$Z(\mathbf{w} \,|\, \mathbf{x}) = \sum_{\mathbf{y}' \in \mathcal{Y}} \exp(\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}') \rangle), \tag{8}$$

and

$$\Phi(\mathbf{x}, \mathbf{y}) = [\sum_{i \in V} \Phi_i(y^{(i)}, \mathbf{x}); \sum_{(i,j) \in E} \Phi_{i,j}(y^{(i)}, y^{(j)}, \mathbf{x})]$$

$$\mathbf{w} = [\mathbf{w}_1; \mathbf{w}_2].$$

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 $R(\mathbf{w} \mid \mathbf{X}, \mathbf{Y})$

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

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

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

# CRFs - 4

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

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

It follows from direct computation that

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

# 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 $\mathcal{E}$

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

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

# That's all

Thanks!