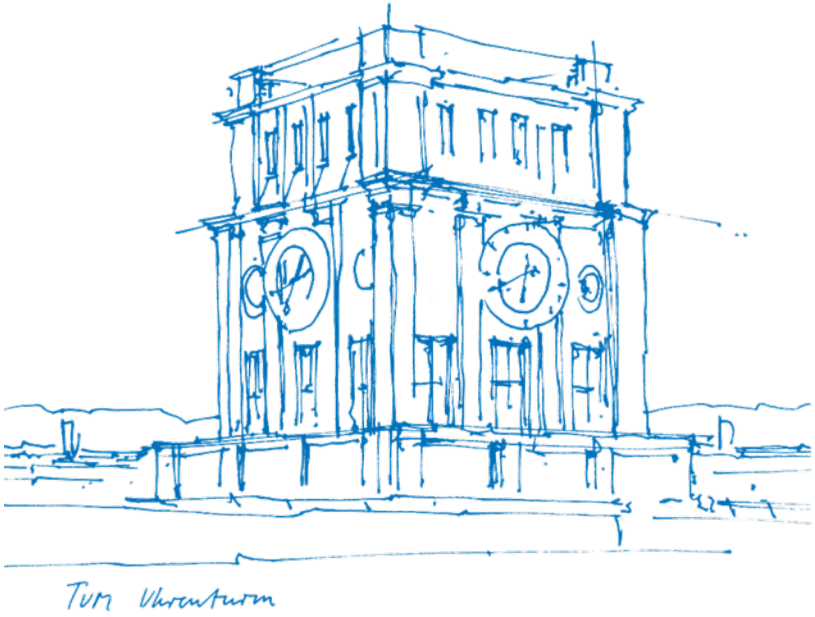
Computer Vision & Artificial Intelligence
Department of Informatics
Technical University of Munich

IV : Learning Graphical Models

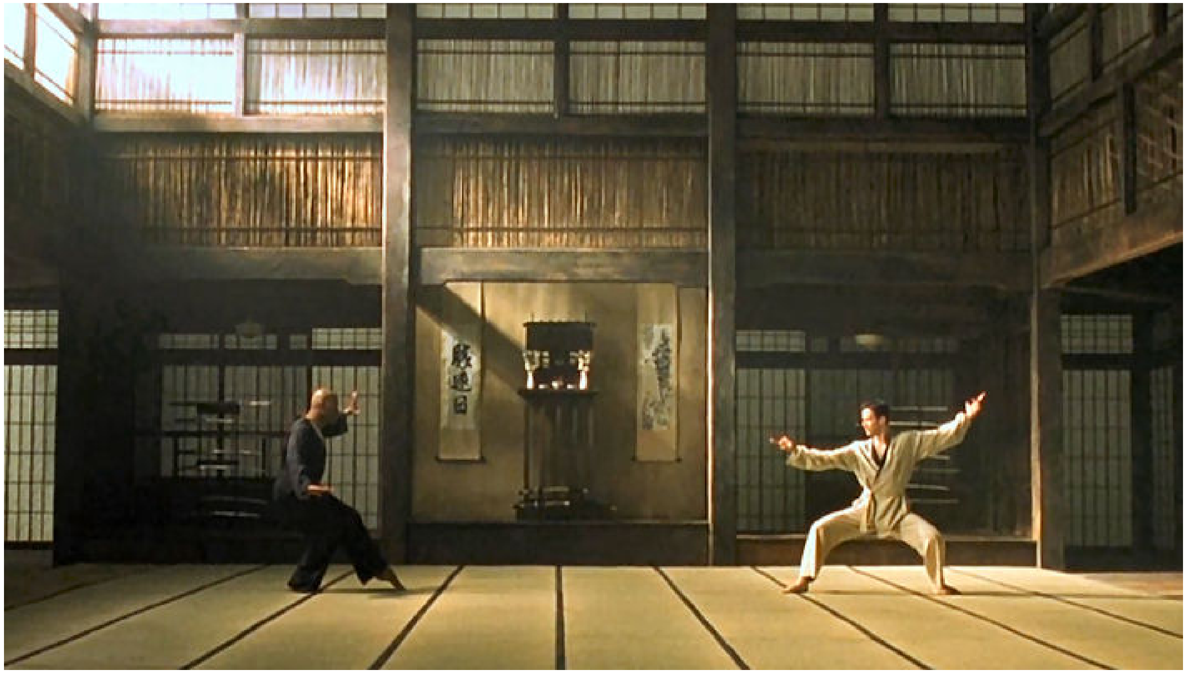
Tao Wu, Yuesong Shen, Zhenzhang Ye

Computer Vision & Artificial Intelligence
Technical University of Munich



Welcome to Learning

- ✓ Graphical Model Representation
- ✓ Inference on Graphical Models
- ⇝ **Learning Graphical Models**



Source: The Matrix (1999).

Goal of Learning

- **Density estimation**: Find p as "close" as possible to the ground-truth distribution r, e.g., in terms of KL divergence (a.k.a. *M-projection*):

$$\min_{\theta} \mathrm{KL}\left(r \mid p(\cdot;\theta)\right).$$

- Specific **prediction** task (e.g. classification, segmentation): Learn a *prediction function*

$$F(x;\theta) = \arg\max_{y} p(y|x;\theta).$$

- The above two goals are both about *parameter learning*. There is another type of learning called **structure/knowledge discovery** — learn the structure of a graphical model (i.e. interaction between random variables).

Maximum Likelihood Estimation

Empirical Distribution and Maximum Likelihood

- In practice, the ground-truth distribution r is assessed via i.i.d. samples $\mathcal{S} = \{x^1, x^2, \dots, x^N\}$ or $\mathcal{S} = \{(x^1, y^1), (x^2, y^2), \dots, (x^N, y^N)\}$.
- That is, r is replaced by an **empirical distribution** of the form

$$r(x) = \frac{1}{|\mathcal{S}|} \sum_{x' \in \mathcal{S}} \delta_{x'}(x), \quad \text{or}$$

$$r(x, y) = \frac{1}{|\mathcal{S}|} \sum_{(x', y') \in \mathcal{S}} \delta_{(x', y')}(x, y).$$

- Density estimation:

$$\arg\min_\theta \mathrm{KL}\left(r \,|\, p(\cdot; \theta)\right) = \arg\min_\theta \mathbb{E}_{x \sim r}[\log r(x)] - \mathbb{E}_{x \sim r}[\log p(x; \theta)]$$

$$= \arg\min_\theta \ell(\theta) := -\mathbb{E}_{x \sim r}[\log p(x; \theta)] = -\frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \log p(x; \theta).$$

We have derived the **maximum likelihood estimation** (MLE). The loss $\ell(\theta)$ is called the *negative log-likelihood* (NLL) loss.

MLE for Learning Bayesian Networks

- Let p be represented by a BN $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, i.e. $p(x;\theta) = \prod_{i\in\mathcal{V}} \theta_i(x_i|x_{\mathrm{Pa}_\mathcal{G}(i)})$, with parameter θ satisfying $\theta_i(x_i|x_{\mathrm{Pa}_\mathcal{G}(i)}) \geq 0$ and $\sum_{x_i} \theta_i(x_i|x_{\mathrm{Pa}_\mathcal{G}(i)}) = 1$.
- MLE for (fully observable) BN $\rightsquigarrow$ minimize the NLL loss $\ell(\theta)$ over θ:

$$\min_\theta \ell(\theta) = -\frac{1}{|\mathcal{S}|}\sum_{x\in\mathcal{S}} \log p(x;\theta) = -\frac{1}{|\mathcal{S}|}\sum_{x\in\mathcal{S}}\sum_{i\in\mathcal{V}} \log \theta_i(x_i|x_{\mathrm{Pa}_\mathcal{G}(i)})$$

$$= -\frac{1}{|\mathcal{S}|}\sum_{x\in\mathcal{S}}\sum_{i\in\mathcal{V}}\sum_{x'_{\{i\}\cup\mathrm{Pa}_\mathcal{G}(i)}} \log \theta_i(x'_i|x'_{\mathrm{Pa}_\mathcal{G}(i)}) \mathbf{1}\{x'_{\{i\}\cup\mathrm{Pa}_\mathcal{G}(i)} = x_{\{i\}\cup\mathrm{Pa}_\mathcal{G}(i)}\}$$

$$= -\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{V}}\sum_{x'_{\{i\}\cup\mathrm{Pa}_\mathcal{G}(i)}} \log \theta_i(x'_i|x'_{\mathrm{Pa}_\mathcal{G}(i)}) \sum_{x\in\mathcal{S}} \mathbf{1}\{x'_{\{i\}\cup\mathrm{Pa}_\mathcal{G}(i)} = x_{\{i\}\cup\mathrm{Pa}_\mathcal{G}(i)}\}$$

$$=: -\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{V}}\sum_{x'_{\{i\}\cup\mathrm{Pa}_\mathcal{G}(i)}} \log \theta_i(x'_i|x'_{\mathrm{Pa}_\mathcal{G}(i)}) N_i(x'_i|x'_{\mathrm{Pa}_\mathcal{G}(i)}),$$

which has a close-form solution: $\theta^*_i(x'_i|x'_{\mathrm{Pa}_\mathcal{G}(i)}) = \dfrac{N_i(x'_i|x'_{\mathrm{Pa}_\mathcal{G}(i)})}{\sum_{x'_i} N_i(x'_i|x'_{\mathrm{Pa}_\mathcal{G}(i)})}$.

Markov Random Field in Log-Linear Form

- Let p be represented by an MRF $\mathcal{H} = (\mathcal{V}, \mathcal{E})$:

$$p(x;\eta) = \frac{1}{Z(\eta)} \prod_{C \in \text{Clique}(\mathcal{H})} \phi_C(x_C;\eta_C),$$

$$Z(\eta) = \sum_x \prod_{C \in \text{Clique}(\mathcal{H})} \phi_C(x_C;\eta_C).$$

- Reparameterize p in the *log-linear form*:

$$p(x;\eta) = \frac{1}{Z(\eta)} \exp\Bigg(\sum_{C \in \text{Clique}(\mathcal{H})} \sum_{x'_C} \mathbf{1}\{x_C = x'_C\} \log \phi_C(x'_C;\eta_C)\Bigg)$$

$$=: \frac{1}{Z(\theta)} \exp(\theta^\top \psi(x)) = p(x;\theta).$$

- $\psi(x)$ is a vector whose entries are given by indicator functions $\mathbf{1}\{x_C = x'_C\}$; θ is a vector whose entries are given by log-potentials $\log \phi_C(x'_C;\eta_C)$.
- More generally, $p(x;\theta)$ of the above form is a member of the **exponential family**; $\psi(x)$ is called the **sufficient statistics**; θ is the **natural parameters**.

MLE for Learning MRFs

- Minimize the NLL loss $\ell(\theta)$ for $p(x;\theta) = \frac{1}{Z(\theta)} \exp(\theta^\top \psi(x))$:

$$\ell(\theta) = -\frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \log p(x;\theta) = -\theta^\top \Big(\frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \psi(x) \Big) + \log Z(\theta),$$

$$\log Z(\theta) = \log \sum_{x} \exp(\theta^\top \psi(x)).$$

- There is no closed form for the optimal solution. Instead, we can derive the gradient of $\ell(\theta)$ as:

$$\nabla_\theta \log Z(\theta) = \sum_{x} \frac{\exp(\theta^\top \psi(x))}{\sum_{x'} \exp(\theta^\top \psi(x'))} \psi(x) = \mathbb{E}_{x \sim p(\cdot;\theta)}[\psi(x)],$$

$$\nabla_\theta \ell(\theta) = \mathbb{E}_{x \sim p(\cdot;\theta)}[\psi(x)] - \mathbb{E}_{x \sim r}[\psi(x)],$$

where $r(x) = \frac{1}{|\mathcal{S}|} \sum_{x' \in \mathcal{S}} \delta_{x'}(x)$ is the empirical distribution.

MLE for Learning MRFs (cont'd)

- We can also derive

$$\begin{aligned}\nabla_\theta^2 \ell(\theta) &= \nabla_\theta^2 \log Z(\theta)\\ &= \mathbb{E}_{x\sim p(\cdot;\theta)}[\psi(x)\psi(x)^\top] - \mathbb{E}_{x\sim p(\cdot;\theta)}[\psi(x)]\,\mathbb{E}_{x\sim p(\cdot;\theta)}[\psi(x)]^\top\\ &= \mathrm{Cov}_{x\sim p(\cdot;\theta)}[\psi(x)]. \qquad \text{(positive semidefinite } \forall\theta)\end{aligned}$$

  This implies that the function $\ell(\theta)$ is *convex* in θ.

- Recall that $\psi(x)$ contains sufficient statistics (or features). A vanishing gradient of the NLL loss

$$\nabla_\theta \ell(\theta) = \mathbb{E}_{x\sim p(\cdot;\theta)}[\psi(x)] - \mathbb{E}_{x\sim r}[\psi(x)] = 0$$

  yields *moment matching* of $\psi(x)$ between "model prediction" and "empirical observation".

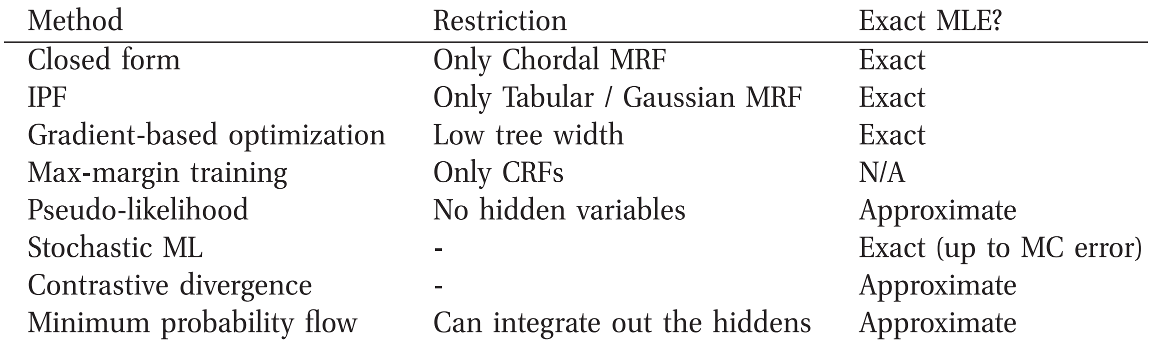
- MLE learning can be numerically carried out by gradient descent iterations:

$$\theta \leftarrow \theta - \tau \nabla_\theta \ell(\theta),$$

  for properly chosen step size τ. Each iteration requires one (approximate) probabilistic inference (e.g. via variational inference or sampling).

Alternatives to Gradient-based Learning

- This lecture focuses on "gradient-based" MLE learning of MRFs/CRFs. It is a general-purpose paradigm but can be computationally expensive.
- There exist various alternatives to gradient-based learning of MRFs (typically effective under more restrictive settings), e.g.:
  - Pseudo-likelihood [Murphy, Section 19.5.4].
  - Iterative proportional fitting (IPF) [Murphy, Section 19.5.7].



Method	Restriction	Exact MLE?
Closed form	Only Chordal MRF	Exact
IPF	Only Tabular / Gaussian MRF	Exact
Gradient-based optimization	Low tree width	Exact
Max-margin training	Only CRFs	N/A
Pseudo-likelihood	No hidden variables	Approximate
Stochastic ML	-	Exact (up to MC error)
Contrastive divergence	-	Approximate
Minimum probability flow	Can integrate out the hiddens	Approximate

Figure: Alternatives to gradient-based learning [Murphy, Table 19.1].

Learning CRFs via Conditional Log-Likelihood

- Consider the prediction function F for a specific prediction task:

$$F(x;\theta) = \arg\max_y p(y|x;\theta),$$

where $p(y|x;\theta)$ is modeled by a conditional random field (CRF):

$$p(y|x;\theta) = \frac{1}{Z(\theta;x)} \exp(\theta^\top \psi(y;x)).$$

- Learn the CRF via the **conditional log-likelihood**:

$$\min_\theta \ell(\theta) = -\frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} \log p(y|x;\theta). \tag{†}$$

- With r_x the marginal distribution of r and $r(\cdot|x)$ the conditional distribution, (†) can be interpreted as an extension of MLE:

$$\min_\theta \mathbb{E}_{x\sim r_x}[\mathrm{KL}\,(r(\cdot|x)\,|\,p(\cdot|x;\theta))].$$

- Conditional log-likelihood learning of CRFs is widely used in supervised learning for classification, segmentation, etc. Note that $p(y|x;\theta)$ also provides confidence of the prediction $y(x) = \arg\max_{y'} p(y'|x;\theta)$.

Learning CRFs by Stochastic Gradient Descent

- Proceed similarly as in MLE for learning MRFs (letting $\mathcal{S}_x = \{x^1, ..., x^N\}$):

$$\ell(\theta) = -\theta^\top \Big(\frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} \psi(y; x) \Big) + \frac{1}{|\mathcal{S}|} \sum_{x\in\mathcal{S}_x} \log Z(\theta; x),$$

$$\log Z(\theta; x) = \log \sum_y \exp(\theta^\top \psi(y; x)).$$

- The gradient and the Hessian of $\ell(\theta)$ can be derived as:

$$\nabla_\theta \log Z(\theta; x) = \sum_y \frac{\exp(\theta^\top \psi(y; x))}{\sum_{y'} \exp(\theta^\top \psi(y'; x))} \psi(y; x) = \mathbb{E}_{y\sim p(\cdot|x;\theta)}[\psi(y; x)],$$

$$\nabla_\theta \ell(\theta) = \mathbb{E}_{x\sim r_x}[\mathbb{E}_{y\sim p(\cdot|x;\theta)}[\psi(y; x)]] - \mathbb{E}_{(x,y)\sim r}[\psi(y; x)],$$

$$\nabla^2_\theta \ell(\theta) = \mathbb{E}_{x\sim r_x}[\mathrm{Cov}_{y\sim p(\cdot|x;\theta)}[\psi(y; x)]].$$

- Note the difference between learning CRFs and learning MRFs. Each $\log Z(\theta; x)$ and its gradient now depend on the data point x. For a large dataset, we often approximate $\mathbb{E}_{x\sim r_x}[\cdot]$ inside $\nabla_\theta \ell(\theta)$ by sampling, leading to a *mini-batch stochastic gradient descent* learning scheme.

Further Reading

- Murphy, Sections 10.4, 19.5.
- Koller & Friedman, Chapters 16, 17, 20.

Learning Latent Variable Models

Latent Variable Models

- We have studied MLE for learning a *fully observable* BN/MRF/CRF. However, full observability is not always the case in practice.
- A **latent variable model** (LVM) refers to a distribution $p(x, z; \theta)$ over two sets of variables x, z, where x are observable from the dataset $\mathcal{S} = \{x^1, x^2, ..., x^N\}$ and z are the latent variables never being observed.
- As an example of LVM, a Gaussian mixture model (GMM) is defined by

$$p(z = k) = \pi_k, \quad (\pi_k \geq 0 \; \forall k, \sum_k \pi_k = 1)$$

$$p(x|z = k) = p_G(x; \mu_k, \Sigma_k) = (2\pi)^{-\frac{n}{2}} |\Sigma_k|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x - \mu_k)^\top \Sigma_k^{-1} (x - \mu_k) \right),$$

$$p(x, z; \{\pi_k, \mu_k, \Sigma_k\}) = p(z)p(x|z) = \sum_k \mathbf{1}\{z = k\} \pi_k p_G(x; \mu_k, \Sigma_k).$$

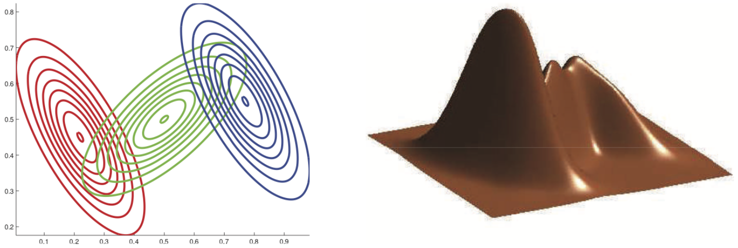


Figure: Mixture of three Gaussians [Murphy, Figure 11.3]. Left: $p(x|z)$; Right: $p(x)$.

MLE for Partially Observable MRFs

We extend gradient-based MLE learning to partially observable MRFs:

$$
\begin{aligned}
p(x, z; \theta) &= \frac{1}{Z(\theta)} \exp\left(\theta^\top \psi(x, z)\right), \\
Z(\theta) &= \sum_{x,z} \exp\left(\theta^\top \psi(x, z)\right). \\
p(x; \theta) &= \sum_z p(x, z; \theta) = \frac{1}{Z(\theta)} \sum_z \exp\left(\theta^\top \psi(x, z)\right), \\
\ell(\theta) &= -\frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \log p(x; \theta) \qquad \qquad (\arg\min_\theta \ell(\theta) \rightsquigarrow \text{MLE}) \\
&= -\frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \log \sum_z \exp\left(\theta^\top \psi(x, z)\right) + \log Z(\theta). \\
\nabla_\theta \ell(\theta) &= -\frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \sum_z \frac{\exp(\theta^\top \psi(x, z))}{\sum_{z'} \exp(\theta^\top \psi(x, z'))} \psi(x, z) + \nabla_\theta \log Z(\theta) \\
&= -\mathbb{E}_{x \sim r}[\mathbb{E}_{z \sim p(\cdot|x;\theta)}[\psi(x, z)]] + \mathbb{E}_{(x,z) \sim p(\cdot,\cdot|\theta)}[\psi(x, z)].
\end{aligned}
$$

MLE for Partially Observable CRFs

(x, y) : observable input/output variables; z : latent variables.

$$
\begin{aligned}
p(y, z|x; \theta) &= \frac{1}{Z(\theta; x)} \exp\left(\theta^\top \psi(y, z; x)\right), \\
Z(\theta; x) &= \sum_{y,z} \exp\left(\theta^\top \psi(y, z; x)\right). \\
p(y|x; \theta) &= \sum_{z} p(y, z|x; \theta) = \frac{1}{Z(\theta; x)} \sum_{z} \exp\left(\theta^\top \psi(y, z; x)\right), \\
\ell(\theta) &= -\frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} \log p(y|x; \theta) \\
&= -\frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} \log \sum_{z} \exp\left(\theta^\top \psi(y, z; x)\right) + \frac{1}{|\mathcal{S}|} \sum_{x\in\mathcal{S}_x} Z(\theta; x). \\
\nabla_\theta\, \ell(\theta) &= -\frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} \sum_{z} \frac{\exp(\theta^\top \psi(y, z; x))}{\sum_{z'} \exp(\theta^\top \psi(y, z'; x))} \psi(y, z; x) + \nabla_\theta \log Z(\theta; x) \\
&= -\mathbb{E}_{(x,y)\sim r}[\mathbb{E}_{z\sim p(\cdot|x,y;\theta)}[\psi(y, z; x)]] + \mathbb{E}_{x\sim r_x}[\mathbb{E}_{(y,z)\sim p(\cdot,\cdot|x;\theta)}[\psi(y, z; x|\theta)]].
\end{aligned}
$$

Expectation Maximization

Expectation maximization (EM) is an important algorithm for learning LVMs, by exploiting the fact that MLE learning for fully observable models is much easier.

EM algorithm:

Require: dataset $\mathcal{S} = \{x^1, x^2, ..., x^N\}$, parameterized distribution $p(x, z; \theta)$.
Initialize θ^0. Iterate $t = 0, 1, 2, ...$ as follows:

1. (E-step) For each $x \in \mathcal{S}$, compute
$$q^t(z|x) := p(z|x; \theta^t).$$
2. (M-step) Compute
$$\theta^{t+1} := \arg\min_{\theta} \widehat{\ell}^t(\theta) = -\frac{1}{|\mathcal{S}|}\sum_{x\in\mathcal{S}}\sum_{z} q^t(z|x)\log p(x, z; \theta).$$

Some remarks:

- Very often, $p(z|x; \theta^t)$ in the E-step has a simple close-form expression.
- The M-step refers to (reweighted) MLE for a fully observable model.

EM for Learning Gaussian Mixture Models

- As a classical example, EM can be applied to learning GMM:

$$\theta = \{\pi_k, \mu_k, \Sigma_k\},$$
$$p(x, z; \theta) = p(z)p(x|z) = \sum_k \mathbf{1}\{z = k\}\pi_k p_G(x; \mu_k, \Sigma_k).$$

- (E-step) $\forall x \in \mathcal{S} : q^t(z = k|x) = p(z = k|x; \theta^t) = \frac{\pi_k^t p_G(x; \mu_k^t, \Sigma_k^t)}{\sum_{k'} \pi_{k'}^t p_G(x; \mu_{k'}^t, \Sigma_{k'}^t)}.$
- (M-step)

$$\theta^{t+1} = \arg\min_\theta \widehat{\ell}^t(\theta) := -\frac{1}{|\mathcal{S}|}\sum_{x\in\mathcal{S}}\sum_z q^t(z|x)\log p(x, z; \theta)$$
$$= -\frac{1}{|\mathcal{S}|}\sum_{x\in\mathcal{S}}\sum_k q^t(z = k|x)\Big(\log \pi_k + \log p_G(x; \mu_k, \Sigma_k)\Big).$$

$\Rightarrow \ \pi_k^{t+1} = \frac{1}{|\mathcal{S}|}\sum_{x\in\mathcal{S}} q^t(z = k|x)$; As the solutions of MLE for Gaussians, $(\mu_k^{t+1}, \Sigma_k^{t+1})$ also has a closed-form solution [Murphy, Section 11.4.2.3].

EM for Learning Partially Observable BNs

- Let p be represented by BN $\mathcal{G} = (\mathcal{V} \cup \mathcal{H}, \mathcal{E})$ with $x = (x_{\mathcal{V}}, x_{\mathcal{H}})$ for observable variables $x_{\mathcal{V}}$ and latent variables $x_{\mathcal{H}}$:

$$p(x_{\mathcal{V}}, x_{\mathcal{H}}; \theta) = \prod_{i \in \mathcal{V} \cup \mathcal{H}} \theta_i(x_i | x_{\mathrm{Pa}_{\mathcal{G}}(i)}).$$

- Denote the empirical observations by $\mathcal{S} = \{x_{\mathcal{V}}^1, x_{\mathcal{V}}^2, \ldots, x_{\mathcal{V}}^N\}$.
- (E-step) $\forall x_{\mathcal{V}} \in \mathcal{S} : q^t(x_{\mathcal{H}} | x_{\mathcal{V}}) := p(x_{\mathcal{H}} | x_{\mathcal{V}}; \theta^t)$.
- (M-step) $\theta^{t+1} := \arg\min_{\theta} \widehat{\ell}^t(\theta)$ with

$$\begin{aligned}
\widehat{\ell}^t(\theta) &= -\frac{1}{|\mathcal{S}|} \sum_{x_{\mathcal{V}} \in \mathcal{S}} \sum_{x_{\mathcal{H}}} q^t(x_{\mathcal{H}} | x_{\mathcal{V}}) \log p(x_{\mathcal{V}}, x_{\mathcal{H}}; \theta) \\
&= -\frac{1}{|\mathcal{S}|} \sum_{x_{\mathcal{V}} \in \mathcal{S}} \sum_{x_{\mathcal{H}}} q^t(x_{\mathcal{H}} | x_{\mathcal{V}}) \sum_{i \in \mathcal{V} \cup \mathcal{H}} \log \theta_i(x_i | x_{\mathrm{Pa}_{\mathcal{G}}(i)}) \\
&= -\frac{1}{|\mathcal{S}|} \sum_{x_{\mathcal{V}} \in \mathcal{S}} \sum_{x_{\mathcal{H}}} q^t(x_{\mathcal{H}} | x_{\mathcal{V}}) \sum_{i \in \mathcal{V} \cup \mathcal{H}} \sum_{x'_{\{i\} \cup \mathrm{Pa}_{\mathcal{G}}(i)}} \log \theta_i(x'_i | x'_{\mathrm{Pa}_{\mathcal{G}}(i)}) \mathbf{1}\{x'_{\{i\} \cup \mathrm{Pa}_{\mathcal{G}}(i)} = x_{\{i\} \cup \mathrm{Pa}_{\mathcal{G}}(i)}\}
\end{aligned}$$

EM for Learning Partially Observable BNs (cont'd)

$$\ldots = -\frac{1}{|\mathcal{S}|}\sum_{x_\mathcal{V}\in\mathcal{S}}\sum_{x_\mathcal{H}} q^t(x_\mathcal{H}|x_\mathcal{V}) \sum_{i\in\mathcal{V}\cup\mathcal{H}}\sum_{x'_{\{i\}\cup\mathrm{Pa}_\mathcal{G}(i)}} \log\theta_i(x'_i|x'_{\mathrm{Pa}_\mathcal{G}(i)})\mathbf{1}\{x'_{\{i\}\cup\mathrm{Pa}_\mathcal{G}(i)} = x_{\{i\}\cup\mathrm{Pa}_\mathcal{G}(i)}\}$$

$$= -\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{V}\cup\mathcal{H}}\sum_{x'_{\{i\}\cup\mathrm{Pa}_\mathcal{G}(i)}} \log\theta_i(x'_i|x'_{\mathrm{Pa}_\mathcal{G}(i)})\sum_{x_\mathcal{V}\in\mathcal{S}}\sum_{x_\mathcal{H}} q^t(x_\mathcal{H}|x_\mathcal{V})\mathbf{1}\{x'_{\{i\}\cup\mathrm{Pa}_\mathcal{G}(i)} = x_{\{i\}\cup\mathrm{Pa}_\mathcal{G}(i)}\}$$

$$=: -\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{V}\cup\mathcal{H}}\sum_{x'_{\{i\}\cup\mathrm{Pa}_\mathcal{G}(i)}} \log\theta_i(x'_i|x'_{\mathrm{Pa}_\mathcal{G}(i)})N_i(x'_i|x'_{\mathrm{Pa}_\mathcal{G}(i)}).$$

- Hence, $\theta^{t+1} := \arg\min_\theta \widehat{\ell}^t(\theta)$ has a closed-form solution:

$$\theta_i(x'_i|x'_{\mathrm{Pa}_\mathcal{G}(i)}) = \frac{N_i(x'_i|x'_{\mathrm{Pa}_\mathcal{G}(i)})}{\sum_{x'_i} N_i(x'_i|x'_{\mathrm{Pa}_\mathcal{G}(i)})}.$$

- In general, evaluation of $q^t(x_\mathcal{H}|x_\mathcal{V})$, hence $N_i(x'_i|x'_{\mathrm{Pa}_\mathcal{G}(i)})$, requires inference.

Convergence Property of EM

- In the following, we study the convergence property of EM.
- Given the empirical distribution r and the model distribution $p(x, z; \theta)$:
  (E-step) $\forall x \in \mathcal{S} : q^t(z|x) = p(z|x; \theta^t)$.
  (M-step) $\theta^{t+1} = \arg\min_\theta \widehat{\ell}^t(\theta) := -\mathbb{E}_{x\sim r}[\mathbb{E}_{z\sim q^t(\cdot|x)}[\log p(x, z|\theta)]]$.
- We derive an *upper bound* for the NLL loss $\ell(\theta)$ by *Jensen's inequality*:

$$
\begin{aligned}
\ell(\theta) := -\mathbb{E}_{x\sim r}[\log p(x;\theta)] &= -\mathbb{E}_{x\sim r}\Big[\log \sum_z p(x,z;\theta)\Big] \\
&= \mathbb{E}_{x\sim r}\Big[-\log \sum_z q(z|x)\frac{p(x,z;\theta)}{q(z|x)}\Big] \\
\text{(Jensen's ineq.)} \quad &\leq \mathbb{E}_{x\sim r}\Big[-\sum_z q(z|x)\log\frac{p(x,z;\theta)}{q(z|x)}\Big] \\
&= \mathbb{E}_{x\sim r}[-\mathbb{E}_{z\sim q(\cdot|x)}[\log p(x,z;\theta)]] + \mathbb{E}_{x\sim r}[\mathbb{E}_{z\sim q(\cdot|x)}[\log q(z|x)]].
\end{aligned}
$$

The Jensen's inequality holds for any $q(\cdot|x)$. It is *tight* if $q(\cdot|x) = p(\cdot|x;\theta)$.

Convergence Property of EM (cont'd)

- Given x and $q(\cdot|x)$, write the upper bound as

$$
\begin{aligned}
L(x; q(\cdot|x), \theta) := -\mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p(x,z;\theta)}{q(z|x)}\right] &= -\mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p(z|x;\theta)p(x|\theta)}{q(z|x)}\right] \\
&= -\mathbb{E}_{z\sim q(\cdot|x)}\left[\log \frac{p(z|x;\theta)}{q(z|x)}\right] - \mathbb{E}_{z\sim q(\cdot|x)}[\log p(x|\theta)] \\
&= \mathrm{KL}\left(q(\cdot|x)\,|\,p(\cdot|x;\theta)\right) - \log p(x|\theta).
\end{aligned}
$$

- E-step ⇝ minimize $L(x; q(\cdot|x), \theta)$ over $q(\cdot|x)$ $\Leftrightarrow$ $q(\cdot|x) = p(\cdot|x;\theta)$.
  M-step ⇝ minimize $\mathbb{E}_{x\sim r}[L(x; q(\cdot|x), \theta)]$ over θ.
  Altogether, EM performs alternating minimization on $\mathbb{E}_{x\sim r}[L(x; q(\cdot|x), \theta)]$.
- Overall, the NLL loss $\ell(\theta^t)$ in EM is monotonically decreasing:

$$
\begin{aligned}
\ell(\theta^{t+1}) &\le \mathbb{E}_{x\sim r}[L(x; q^t(\cdot|x), \theta^{t+1})] && \text{(Jensen)} \\
&\le \mathbb{E}_{x\sim r}[L(x; q^t(\cdot|x), \theta^{t})] && \text{(M-step)} \\
&= \ell(\theta^t). && \text{(E-step makes Jensen tight)}
\end{aligned}
$$

- In practice, EM typically converges to a *local minimizer* of the NLL loss.

Further Reading

- Murphy, Sections 11.4, 19.5.
- Koller & Friedman, Chapter 19.

Structured Support Vector Machine

Structured Risk Minimization

- Let $p(y|x;\theta)$ be modeled by a CRF, i.e., $p(y|x;\theta) = \frac{1}{Z(\theta;x)} \exp(\theta^\top \psi(y;x))$.
- Consider the prediction function F:

$$F(x;\theta) = \arg\max_y p(y|x;\theta) = \arg\max_y \theta^\top \psi(y;x).$$

- Previously, we employed the maximum conditional log-likelihood estimation to learn parameter θ (let $\mathcal{S} = \{(x^1, y^1), ..., (x^N, y^N)\}$):

$$\min_\theta -\frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} \log p(y|x;\theta).$$

- We introduce another approach called the **structured risk minimization**:

$$\min_\theta R(\theta) + \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} \Delta(y, F(x;\theta)).$$

  - Δ is a loss such that $\Delta(y,y) = 0,\ \Delta(y,y') \geq 0\ \forall y, y'$; e.g. the *0-1 loss* $\Delta(y,y') = \mathbf{1}\{y \neq y'\}$.
  - R is a convex *regularizer* on θ (to avoid overfitting); e.g. $R(\theta) = \frac{1}{2\sigma}\|\theta\|^2$.

Structured Support Vector Machine

- Pros and cons of structured risk minimization:

$(+)$ Directly minimizes the "expected loss" of interest.

$(-)$ $F(x;\theta) = \arg\max_y \theta^\top \psi(y;x)$ provides no probabilistic interpretation of y.

$(+)$ Evaluation of $F(x;\theta)$ benefits from fast MAP inference.

$(-^*)$ The loss $\Delta(y,\cdot)$ is discontinuous, hence difficult to optimize.

- Now we introduce **structured support vector machine** (SSVM):

$$\min_\theta \ell_{\text{SSVM}}(\theta) := R(\theta) + \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} \max_{y'}\{\Delta(y,y') + \theta^\top\psi(y';x) - \theta^\top\psi(y;x)\}.$$

- SSVM provides a *convex upper bound* of the loss Δ:

$$\begin{aligned}\Delta(y,F(x;\theta)) &\leq \Delta(y,F(x;\theta)) + \theta^\top\psi(F(x;\theta);x) - \theta^\top\psi(y;x)\\ &\leq \max_{y'}\{\Delta(y,y') + \theta^\top\psi(y';x) - \theta^\top\psi(y;x)\}.\end{aligned}$$

The last expression is a convex function of θ because it is the pointwise maximum of a set of affine (in particular convex) functions of θ.

Connection to Classical SVM

- Naturally, SSVM can be specialized to classical SVM. Assume that
  - Binary-valued $y \in \mathcal{Y} = \{+1, -1\}$);
  - 0-1 loss $\Delta(y, y') = \mathbf{1}\{y \neq y'\}$;
  - Sufficient statistics $\psi(y; x) = \frac{1}{2} yx$.
- This implies binary linear SVM formulation:

$$F(x; \theta) = \arg\max_{y} \theta^\top \psi(y; x) = \operatorname{sgn}(\theta^\top x).$$

$$\Delta(y, y') + \theta^\top \psi(y'; x) - \theta^\top \psi(y; x) = \begin{cases} 0 & \text{if } y = y', \\ 1 - y\theta^\top x & \text{if } y \neq y'. \end{cases}$$

$$\min_{\theta} \ell_{\text{SVM}}(\theta) := R(\theta) + \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} \underbrace{\max\{0, 1 - y\theta^\top x\}}_{\text{"hinge loss"}}.$$

Training SSVM by Subgradient Descent

```
   Require: initial step size $\tau > 0$, maximal iteration number $T$.
0. Initialize $\theta^0 := 0$.
   for $t \in \{0, 1, 2, ..., T\}$ do
     for $(x, y) \in \mathcal{S}$ do
1.       Compute $\widehat{y}^t(x) := \arg\max_{y'} \Delta(y, y') + (\theta^t)^\top \psi(y'; x)$.
     end for
2.   Compute $\delta\theta^t := \nabla R(\theta^t) + \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} \big(\psi(\widehat{y}^t(x); x) - \psi(y; x)\big)$.
3.   Compute $\theta^{t+1} := \theta^t - \frac{\tau}{t+1} \delta\theta^t$.
   end for
```

Some remarks:

- Step 1 finds the *active branch* of $\max_{y'}\{\Delta(y, y') + (\theta^t)^\top \psi(y'; x) - \theta^\top \psi(y; x)\}$.
- In Step 2, $\delta\theta^t$ is a *subgradient* of the objective ℓ_{SSVM} at θ^t.
- For efficiency, $\mathcal{S}$ in Step 2 can be replaced by a random mini-batch of $\mathcal{S}$.
- The scheduling of step sizes $\{\frac{\tau}{t+1}\}_{t=0}^{\infty}$ is standard for subgradient descent.

Latent SSVM

- SSVM can be extended to learn partially observable CRFs. Consider

$$p(y, z|x; \theta) = \frac{1}{Z(\theta; x)} \exp\left(\theta^\top \psi(y, z; x)\right).$$

$$F(x; \theta) = \arg\max_y \left(\max_z p(y, z|x; \theta)\right) = \arg\max_y \left(\max_z \theta^\top \psi(y, z; x)\right).$$

- **Latent SSVM**:

$$\min_\theta \ell_{\text{l-SSVM}}(\theta) := R(\theta) + \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} \Bigg(\max_{y'}\{\Delta(y, y') + \max_z \theta^\top \psi(y', z; x)\} - \max_z \theta^\top \psi(y, z; x)\Bigg).$$

- Different from SSVM, $\ell_{\text{l-SSVM}}(\theta)$ is no longer convex in θ. In fact, it admits a special structure called "difference of convex functions", i.e., $\ell_{\text{l-SSVM}}(\theta) =: f(\theta) - g(\theta)$ for two convex functions f, g.
- Numerical optimization of $\ell_{\text{l-SSVM}}(\theta)$ can be carried out by an algorithm called *concave-convex procedure* (CCCP) [Murphy, Algorithm 19.5]. This algorithm is another example of majorize-minimize algorithms (same for EM).

Further Reading

- Murphy, Section 19.7.
- Nowozin & Lampert, Section 6.