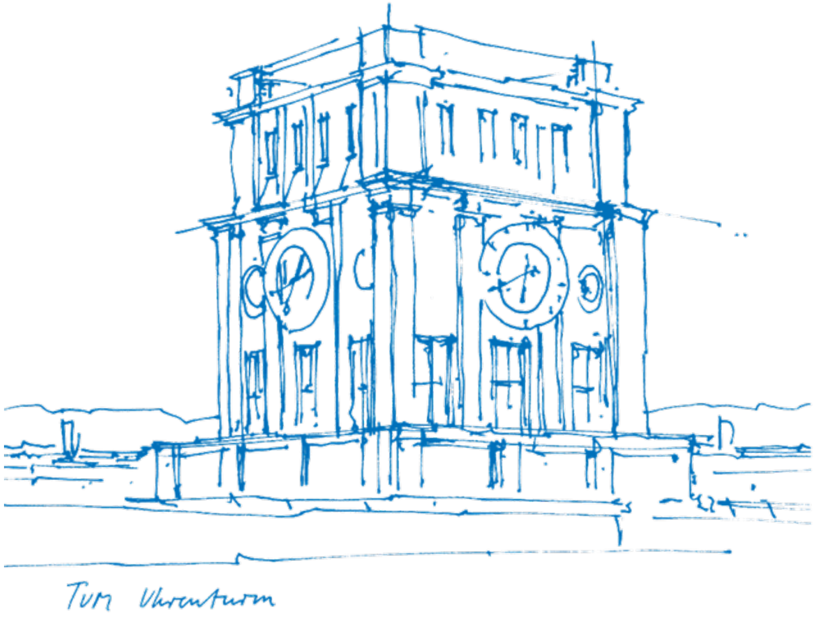
IV : Learning Graphical Models

Tao Wu, Yuesong Shen, Zhenzhang Ye

Computer Vision & Artificial Intelligence
Technical University of Munich



Goal of Learning

So far in the lecture:

- Graphical Model Representation
- Inference on Graphical Models

$\rightsquigarrow$ **Learning Graphical Models**

Goal of learning:

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

$$\min_{\theta} \mathrm{KL}\left(q \,|\, p(\cdot;\theta)\right).$$

- Specific **prediction** task (e.g. classification, segmentation): Learn a prediction function $F(x;\theta) := \arg\max_y p(y|x;\theta)$.
- **Structure/Knowledge discovery**: Learn the structure of a graphical model (i.e. interaction between random variables).

Maximum Likelihood Estimation

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

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

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

- Density estimation:

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

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

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 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$:

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

with parameter θ satisfying $\theta(x_i | x_{\mathrm{Pa}_{\mathcal{G}}(i)}) \geq 0$ and $\sum_{x_i} \theta(x_i | x_{\mathrm{Pa}_{\mathcal{G}}(i)}) = 1$.

- MLE for (fully observable) BN $\rightsquigarrow$ minimize the NLL loss $\ell(\theta)$ over θ:

$$\begin{aligned}\min_\theta \ell(\theta) &= -\frac{1}{|\mathcal{S}|} \sum_{x' \in \mathcal{S}} \log p(x';\theta) = -\frac{1}{|\mathcal{S}|} \sum_{x' \in \mathcal{S}} \sum_{i \in \mathcal{V}} \log \theta(x'_i | x'_{\mathrm{Pa}_{\mathcal{G}}(i)}) \\ &= -\frac{1}{|\mathcal{S}|} \sum_{x' \in \mathcal{S}} \sum_{i \in \mathcal{V}} \sum_{x_{\{i\} \cup \mathrm{Pa}_{\mathcal{G}}(i)}} \log \theta(x_i | x_{\mathrm{Pa}_{\mathcal{G}}(i)}) \mathbf{1}\{x_{\{i\} \cup \mathrm{Pa}_{\mathcal{G}}(i)} = x'_{\{i\} \cup \mathrm{Pa}_{\mathcal{G}}(i)}\} \\ &= -\frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{V}} \sum_{x_{\{i\} \cup \mathrm{Pa}_{\mathcal{G}}(i)}} \log \theta(x_i | x_{\mathrm{Pa}_{\mathcal{G}}(i)}) \Big(\sum_{x' \in \mathcal{S}} \mathbf{1}\{x_{\{i\} \cup \mathrm{Pa}_{\mathcal{G}}(i)} = x'_{\{i\} \cup \mathrm{Pa}_{\mathcal{G}}(i)}\} \Big),\end{aligned}$$

which has a close-form solution:

$$\theta^*(x_i | x_{\mathrm{Pa}_{\mathcal{G}}(i)}) = \frac{\sum_{x' \in \mathcal{S}} \mathbf{1}\{x_{\{i\} \cup \mathrm{Pa}_{\mathcal{G}}(i)} = x'_{\{i\} \cup \mathrm{Pa}_{\mathcal{G}}(i)}\}}{\sum_{x' \in \mathcal{S}} \mathbf{1}\{x_{\mathrm{Pa}_{\mathcal{G}}(i)} = x'_{\mathrm{Pa}_{\mathcal{G}}(i)}\}} = \frac{\#(x'_i, x'_{\mathrm{Pa}_{\mathcal{G}}(i)})}{\#(x'_{\mathrm{Pa}_{\mathcal{G}}(i)})}, \quad \forall i \in \mathcal{V}.$$

Learning MRFs in Log-Linear Form

- Let p be represented by a 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-energies $\log \phi_C(x'_C; \eta_C)$.
- 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 Markov Random Fields

- 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:

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

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

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

MLE for Learning Markov Random Fields (cont'd)

- We can also derive (exercise!)

$$\begin{aligned}\nabla_\theta^2 \ell(\theta) &= \nabla_\theta^2 \log Z(\theta) \\ &= \mathbb{E}_{x\sim p(\cdot;\theta)}[\psi(x)\psi(x)^\top] - \mathbb{E}_{x\sim p(\cdot;\theta)}[\psi(x)]\,\mathbb{E}_{x\sim p(\cdot;\theta)}[\psi(x)]^\top \\ &= \mathrm{Cov}_{x\sim p(\cdot;\theta)}[\psi(x)] \quad (\geq 0 \ \ \forall\theta).\end{aligned}$$

  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

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

  implies *moment matching* of $\psi(x)$ between model prediction and empirical distribution.

- 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).

Conditional Log-Likelihood for Learning CRFs

- Consider the prediction function in 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**:

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

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

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

- 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

- Proceed similarly as in MLE for learning MRFs (letting $\mathcal{S}_x = \bigcup_{(x,y)\in\mathcal{S}}\{x\}$):

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

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

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

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

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

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

- 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 have to approximate $\mathbb{E}_{x\sim q_x}[\cdot]$ 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.