Statistical Machine Learning (BE4M33SSU)
Lecture 5.

Czech Technical University in Prague

- Unsupervised Learning
- Maximum Likelihood Estimator, consistency
- Expectation Maximisation Algorithm
- Examples

5.1 Why do we need unsupervised learning?

If the model $p(x,y)$ is known $\Rightarrow$ $h(x) = \arg\max_{y \in \mathcal{Y}} \sum_{y' \in \mathcal{Y}} p(x,y')\ell(y',y)$

Learning so far: $p(x,y)$ unknown

Given: hypothesis class $\mathcal{H}$ and i.i.d. training data $\mathcal{T}^m = \{(x^i, y^i) \mid i = 1,2,\ldots,m\}$

ERM: $h = \arg\min_{h \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \ell(y^i, h(x^i))$

Learning now: training data possibly incomplete (missing information)

Given: model class $p_\theta(x,y)$, $\theta \in \Theta$, but the true value θ_0 is unknown

Training data i.i.d. generated from p_{θ_0}, e.g.

1. $\mathcal{T}^m = \{(x^i, y^i) \mid i = 1,2,\ldots,m\}$ as before,
2. $\mathcal{T}^m = \{x^i \mid i = 1,2,\ldots,m\}$
3. $Z = f(X,Y)$ is a random variable, $\mathcal{T}^m = \{z^i \mid i = 1,2,\ldots,m\}$

or, combinations thereof.

5.1 Why do we need unsupervised learning?

Approach:

1. use the Maximum Likelihood estimate $\theta^* = \arg\max_{\theta \in \Theta} \log p_\theta(\mathcal{T}^m)$,
2. and the predictor $h(x) = \arg\min_{y \in \mathcal{Y}} \sum_{y' \in \mathcal{Y}} p_{\theta^*}(x, y') \ell(y', y)$.

Questions:

- Is the Maximum Likelihood estimator $\theta^*(\mathcal{T}^m)$ consistent? I.e., does

$$\mathbb{P}_{\theta_0}\big(\|\theta^*(\mathcal{T}^m) - \theta_0\| > \epsilon\big) \xrightarrow{m \to \infty} 0$$

  hold for any $\epsilon > 0$?

- How to implement the estimator in case of training data with missing information (unsupervised learning)?

5.2 Consistency of the Maximum Likelihood estimator

$\mathcal{T}^m = \{z^i \mid i = 1, \ldots, m\}$ i.i.d. generated from $p_{\theta_0}(z)$, $\theta_0 \in \Theta$ unknown

Which conditions ensure consistency of the MLE $\theta^* = \arg\max_{\theta \in \Theta} \log p_\theta(\mathcal{T}^m)$?

---

log-likelihood of training data $L(\theta, \mathcal{T}^m) := \frac{1}{m} \sum_{i=1}^{m} \log p_\theta(z_i)$

expected log-likelihood $L(\theta) = \mathbb{E}_{\theta_0}\big(L(\theta, \mathcal{T}^m)\big) = \sum_{z \in \mathcal{Z}} p_{\theta_0}(z) \log p_\theta(z)$

How to check consistency of MLE (main steps):

- prove that $\theta_0 = \arg\max_{\theta \in \Theta} L(\theta)$ holds, i.e. the model is identifiable
- ensure that the Uniform Law of Large Numbers (ULLN) holds, i.e.

$$\mathbb{P}_{\theta_0}\big(\sup_{\theta \in \Theta} |L(\theta, \mathcal{T}^m) - L(\theta)| > \epsilon\big) \xrightarrow{m \to \infty} 0$$

holds for any $\epsilon > 0$.

5.2 Consistency of the Maximum Likelihood estimator

The first condition, i.e. identifiability of the model θ_0 is easy to prove if $p_{\theta_0}(z) \not\equiv p_\theta(z)$ holds $\forall \theta \neq \theta_0$.

Let $p(z), q(z)$ be two probability distributions s.t. $p \not\equiv q$. Then

$$\sum_{z \in \mathcal{Z}} p(z) \log p(z) > \sum_{z \in \mathcal{Z}} p(z) \log q(z).$$

This follows from strict concavity of the function $\log(x)$:

$$\sum_{z \in \mathcal{Z}} p(z) \log \frac{q(z)}{p(z)} < \log \sum_{z \in \mathcal{Z}} \frac{q(z)p(z)}{p(z)} = \log 1 = 0$$

Recall the Kullback-Leibler divergence for distributions

$$D_{KL}(p \| q) = \sum_{z \in \mathcal{Z}} p(z) \log \frac{p(z)}{q(z)}$$

5.2 Consistency of the Maximum Likelihood estimator

Proving the second condition, i.e. ULLN directly, is sometimes not too complicated (see seminar).

Sufficient conditions that ensure the ULLN:

- $\Theta \subset \mathbb{R}^k$ is compact, $L(\theta, \mathcal{T}^m)$ is continuous in θ and there is a function $d(z) \geqslant \log p_\theta(z)$ $\forall \theta$ with $\mathbb{E}_{\theta_0}(d(z)) < \infty$,
- $\log p_\theta(z)$ is a concave function of θ, $\Theta \subset \mathbb{R}^k$ is convex and $\theta_0 \in \mathrm{int}(\Theta)$.

5.3 The Expectation Maximisation Algorithm

Model class $p_\theta(x,y)$, $\theta \in \Theta$, but the true value θ_0 is unknown

Training data $\mathcal{T}^m = \{x^i \mid i = 1, 2, \ldots, m\}$ i.i.d. generated from p_{θ_0}

How shall we implement the MLE

$$\theta^*(\mathcal{T}^m) = \arg\max_{\theta\in\Theta} \frac{1}{m}\sum_{i=1}^{m} \log p_\theta(x^i) = \arg\max_{\theta\in\Theta} \frac{1}{m}\sum_{i=1}^{m} \log \sum_{y\in\mathcal{Y}} p_\theta(x^i, y)$$

Expectation Maximisation Algorithm (Schlesinger, 1968, Sundberg, 1974, Dempster, Laird, and Rubin, 1977)

5.3 The Expectation Maximisation Algorithm

Schlesinger (1968): Introduce arbitrary numbers $\alpha(y \mid x^i) \geqslant 0$, for each $x^i \in \mathcal{T}^m$, s.t. $\sum_{y \in \mathcal{Y}} \alpha(y \mid x^i) = 1$. Write the log-likelihood as

$$L(\theta, \mathcal{T}^m) = \frac{1}{m} \sum_{i=1}^{m} \log \sum_{y \in \mathcal{Y}} p_\theta(x^i, y) =$$

$$= \frac{1}{m} \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \alpha(y \mid x^i) \log p_\theta(x^i, y) - \frac{1}{m} \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \alpha(y \mid x^i) \log \underbrace{\frac{p_\theta(x^i, y)}{\sum_{y' \in \mathcal{Y}} p_\theta(x^i, y')}}_{p_\theta(y \mid x^i)}$$

Initialise the algorithm with $\theta^{(0)}$ and iterate (until convergence in α)

E-step Set the auxiliary variables to $\alpha^{(t)}(y \mid x^i) = p_{\theta^{(t)}}(y \mid x^i)$

M-step Solve the Maximum Likelihood estimation for complete training data

$$\theta^{(t+1)} = \arg\max_{\theta \in \Theta} \frac{1}{m} \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \alpha^{(t)}(y \mid x^i) \log p_\theta(x^i, y)$$

Claim: the sequence $L(\theta^{(t)}, \mathcal{T}^m)$, $t = 0, 1, \ldots$ is non-decreasing, the sequence $\alpha^{(t)}$ converges.

5.3 The Expectation Maximisation Algorithm

Minka (1998): Consider the following lower bound of the log-likelihood

$$L(\theta, \mathcal{T}^m) = \frac{1}{m}\sum_{i=1}^{m} \log \sum_{y\in\mathcal{Y}} p_\theta(x^i, y) = \frac{1}{m}\sum_{i=1}^{m} \log \sum_{y\in\mathcal{Y}} \frac{\alpha(y \mid x^i)}{\alpha(y \mid x^i)} p_\theta(x^i, y) \geqslant$$

$$L_B(\theta, \mathcal{T}^m) = \frac{1}{m}\sum_{i=1}^{m}\sum_{y\in\mathcal{Y}} \alpha(y \mid x^i) \log p_\theta(x^i, y) - \frac{1}{m}\sum_{i=1}^{m}\sum_{y\in\mathcal{Y}} \alpha(y \mid x^i) \log \alpha(y \mid x^i)$$

Maximise L_B by block-coordinate ascent, i.e. start with some $\theta^{(0)}$ and iterate

E-step Maximisation w.r.t. α-s gives $\alpha^{(t)}(y \mid x^i) = p_{\theta^{(t)}}(y \mid x^i)$

M-step maximisation w.r.t. θ means to solve the MLE for complete training data

$$\theta^{(t+1)} = \arg\max_{\theta\in\Theta} \frac{1}{m}\sum_{i=1}^{m}\sum_{y\in\mathcal{Y}} \alpha^{(t)}(y \mid x^i) \log p_\theta(x^i, y)$$

Claims:

- The bound is tight if $\alpha(y \mid x^i) = p_\theta(y \mid x^i)$,
- see previous slide

5.3 The Expectation Maximisation Algorithm

Compare Schlesinger's representation of L and Minka's lower bound L_B

$$L(\theta,\alpha,\mathcal{T}^m) = \frac{1}{m}\sum_{i=1}^{m}\sum_{y\in\mathcal{Y}}\alpha(y\mid x^i)\log p_\theta(x^i,y) - \frac{1}{m}\sum_{i=1}^{m}\sum_{y\in\mathcal{Y}}\alpha(y\mid x^i)\log p_\theta(y\mid x^i)$$

$$L_B(\theta,\alpha,\mathcal{T}^m) = \frac{1}{m}\sum_{i=1}^{m}\sum_{y\in\mathcal{Y}}\alpha(y\mid x^i)\log p_\theta(x^i,y) - \frac{1}{m}\sum_{i=1}^{m}\sum_{y\in\mathcal{Y}}\alpha(y\mid x^i)\log \alpha(y\mid x^i)$$

5.4 Example: Exponential Families

Exponential family for observations $x \in \mathcal{X}$ and hidden labels $y \in \mathcal{Y}$

$$p_{\boldsymbol{u}}(x, y) = \frac{1}{Z(\boldsymbol{u})} \exp \langle \boldsymbol{\phi}(x, y), \boldsymbol{u} \rangle$$

where

- $\boldsymbol{\phi}\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^k$ is a generalised feature map,
- $\boldsymbol{u} \in \mathbb{R}^k$ is a parameter vector and
- $Z(\boldsymbol{u})$ is the normalisation constant $Z(\boldsymbol{u}) = \sum_{x,y} \exp \langle \boldsymbol{\phi}(x, y), \boldsymbol{u} \rangle$

Supervised learning:

(1) Each model of the class is identifiable under mild conditions (see Assignment 2 of Seminar 3)

(2) $\log p_{\boldsymbol{u}}(x, y)$ is a concave function of $\boldsymbol{u}$, hence ULLN holds for exponential families

$$\log p_{\boldsymbol{u}}(x, y) = \langle \boldsymbol{\phi}(x, y), \boldsymbol{u} \rangle - \log Z(\boldsymbol{u})$$

5.4 Example: Exponential Families

Computing the second derivative of $\log Z(\boldsymbol{u})$

$$\nabla_{\boldsymbol{u}} \log Z(\boldsymbol{u}) = \mathbb{E}_{\boldsymbol{u}} \boldsymbol{\phi}$$
$$\nabla^2_{\boldsymbol{u}} \log Z(\boldsymbol{u}) = \mathbb{E}_{\boldsymbol{u}} \big[(\boldsymbol{\phi} - \mathbb{E}_{\boldsymbol{u}} \boldsymbol{\phi}) \otimes (\boldsymbol{\phi} - \mathbb{E}_{\boldsymbol{u}} \boldsymbol{\phi}) \big]$$

The expectation of a positive semi-definite (random) matrix is positive semi-definite. Hence, $\log Z(\boldsymbol{u})$ is convex. Consequently, the ULLN holds for the ML estimator.

(3) Learning task: Given training data $\mathcal{T}^m = \big\{ (x^i, y^i) \bigm| i = 1, 2, \ldots, m \big\}$, the MLE reads

$$L(\boldsymbol{u}, \mathcal{T}^m) = \frac{1}{m} \sum_{i=1}^{m} \langle \boldsymbol{\phi}(x^i, y^i), \boldsymbol{u} \rangle - \log Z(\boldsymbol{u}) = \Big\langle \overline{\boldsymbol{\Phi}}^m, \boldsymbol{u} \Big\rangle - \log Z(\boldsymbol{u}) \to \max_{\boldsymbol{u}}$$

The objective function is concave in $\boldsymbol{u}$. Apply some convex minimisation algorithm (provided that computation of $\log Z(\boldsymbol{u})$ is tractable).

5.4 Example: Exponential Families

Unsupervised learning: Given training data $\mathcal{T}^m = \{x^i \mid i = 1, \ldots, m\}$, the MLE task reads

$$L(\boldsymbol{u}, \mathcal{T}^m) = \frac{1}{m} \sum_{i=1}^{m} \log \sum_{y \in \mathcal{Y}} p_{\boldsymbol{u}}(x^i, y) \rightarrow \max_{\boldsymbol{u}}$$

Recall the EM algorithm: Maximise Minka's lower bound $L_B(\theta, \alpha, \mathcal{T}^m)$ of the log-likelihood by block-coordinate ascent, i.e., start with some $\boldsymbol{u}^{(0)}$ and iterate

E-step Maximisation w.r.t. α-s for fixed $\boldsymbol{u}^{(t)}$ gives

$$\alpha^{(t)}(y \mid x^i) = p_{\boldsymbol{u}^{(t)}}(y \mid x^i) = \frac{\exp \left\langle \boldsymbol{\phi}(x^i, y), \boldsymbol{u}^{(t)} \right\rangle}{\sum_{y' \in \mathcal{Y}} \exp \left\langle \boldsymbol{\phi}(x^i, y'), \boldsymbol{u}^{(t)} \right\rangle}$$

M-step Maximisation w.r.t. $\boldsymbol{u}$ for fixed $\alpha^{(t)}$ reads

$$\frac{1}{m} \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \alpha^{(t)}(y \mid x^i) \langle \boldsymbol{\phi}(x^i, y), \boldsymbol{u} \rangle - \log Z(\boldsymbol{u}) \rightarrow \max_{\boldsymbol{u}}$$

5.4 Example: Exponential Families

Denoting

$$\mathbf{\Phi}^{(t)} = \frac{1}{m} \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \alpha^{(t)}(y \mid x^i) \boldsymbol{\phi}(x^i, y),$$

we get the same type of optimisation task as for supervised learning!

$$\left\langle \mathbf{\Phi}^{(t)}, \boldsymbol{u} \right\rangle - \log Z(\boldsymbol{u}) \to \max_{\boldsymbol{u}} .$$

Additional reading:

Schlesinger, Hlavac, Ten Lectures on Statistical and Structural Pattern Recognition, Chapter 6, Kluwer 2002 (also available in Czech, e.g. in CMP library)

Thomas P. Minka, Expectation-Maximization as lower bound maximization, 1998 (short tutorial, available in internet)