Statistical Machine Learning (BE4M33SSU)
Lecture 8: Generative learning, EM-Algorithm

Czech Technical University in Prague

- Why do we need generative learning?
- Tools & Ingredients
- Maximum Likelihood Estimator, consistency
- Expectation Maximisation Algorithm

1. Why do we need generative learning?

Discriminative learning: $p(x,y)$ unknown

- define a hypothesis class $\mathcal{H}$ of predictors $h\colon \mathcal{X} \to \mathcal{Y}$,
- given a training set $\mathcal{T}^m$, learn h_m by empirical risk minimisation.

However:

- what if we need the uncertainty of the prediction $h_m(x)$?
- how to learn the predictor if only a part of the training data is annotated?
- what if the statistical relation between x and y depends on some *latent* variables z, which we can not observe in principle? I.e. $p(x,y,z)$, but we never see z in the training data.
- what if we want to learn models that can generate realistic data x?

1. Why do we need generative learning?

Generative learning:

- Try to model the unknown distribution $p(x, y)$ and estimate it from training data $\mathcal{T}^m$ $\Rightarrow p_m(x, y)$.
- Then predict by

$$h(x) = \arg\min_{y \in \mathcal{Y}} \sum_{y' \in \mathcal{Y}} p_m(y' \,|\, x)\, \ell(y', y).$$

- uncertainty of the prediction can be obtained from $p_m(y \,|\, x)$,
- data can be generated from $p_m(x \,|\, y)$.

When trying to estimate $p_m(x, y)$, we need to restrict the search to some finite or infinite set of distributions.

We also need similarity measure(s) for distributions.

2. Tools & ingredients

Parametrised distribution family: A set of distributions with common structure, defined up to unknown parameters.

Example 1. Set of all multivariate normal distributions $\mathcal{N}(\mu, V)$ on $\mathbb{R}^n$

$$p_{\mu,V}(x) = \frac{1}{(2\pi)^{n/2}|V|^{1/2}} \exp\Big[-\frac{1}{2}(x-\mu)\cdot V^{-1}\cdot(x-\mu)\Big]$$

parametrised by $\mu \in \mathbb{R}^n$ and a positive (semi) definite $m \times m$ matrix V.

Example 2. An *exponential family* with density

$$p_\theta(x) = \exp\big[\langle\phi(x), \theta\rangle - A(\theta)\big],$$

where

$\phi(x) \in \mathbb{R}^n$ is the sufficient statistic,

$\theta \in \mathbb{R}^n$ is the (natural) parameter and

$A(\theta)$ is the cumulant function defined by

$$A(\theta) = \log \int_{\mathcal{X}} \exp\big[\langle\phi(x), \theta\rangle\big]\, dx$$

2. Tools & ingredients

Kullback-Leibler divergence: Similarity of distributions $p(x)$ and $q(x)$:

$$D_{KL}(p(x) \parallel q(x)) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}$$

D_{KL} is non-negative and is zero if and only if $p(x) = q(x)\ \forall x \in \mathcal{X}$. This follows from strict concavity of the function $\log(x)$

$$-D_{KL}(p \parallel q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{q(x)}{p(x)} \leqslant \log \sum_{x \in \mathcal{X}} \frac{q(x)p(x)}{p(x)} = \log 1 = 0$$

- D_{KL} can be generalised for continuous distributions.
- it becomes ∞ if the support of p is not contained in the support of q.

3. Maximum Likelihood estimator, consistency

Given: a parametrised family of distributions $p_\theta(x, y)$, $\theta \in \Theta$ and an i.i.d. training set $\mathcal{T}^m = \{(x^j, y^j) \in \mathcal{X} \times \mathcal{Y} \mid j = 1, \ldots, m\}$ generated from $p_{\theta_0}(x, y)$ with unknown θ_0.

Task: estimate θ_0

Maximum likelihood estimator: estimate θ_0 by maximising the joint probability (density) of the training set w.r.t. θ

$$\theta_m \in \arg\max_{\theta \in \Theta} \sum_{j=1}^{m} \log p_\theta(x^j, y^j)$$

Notice that θ_m depends on $\mathcal{T}^m$, thus it is a random variable. MLE has following properties

- MLE can be biased, however
- MLE is asymptotically consistent, i.e. the sequence θ_m, $m \to \infty$ converges in probability to θ_0
- MLE has lowest possible variance (MSE) among all consistent estimators.

3. Maximum Likelihood estimator, consistency

Example 3. (Gaussian Discriminative Analysis)

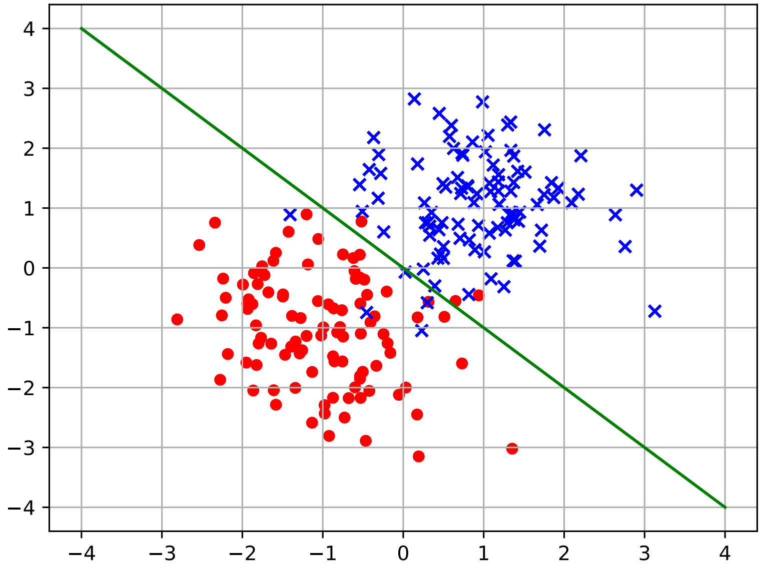
$x \in \mathbb{R}^n$, $y \in \{0, 1\}$ with $y \sim Ber(\alpha)$ and $x \mid y \sim \mathcal{N}(\mu_y, V)$, i.e.

$$p(y) = \alpha^y (1-\alpha)^{1-y}$$
$$p(x \mid y) = \frac{1}{(2\pi)^{n/2} |V|^{1/2}} \exp\Big[-\frac{1}{2}(x-\mu_y)^T V^{-1} (x-\mu_y)\Big]$$

MLE for training data $\mathcal{T}^m = \big\{(x^j, y^j) \mid j = 1, \ldots, m\big\}$:

Denote $I_1 = \{j \mid y^j = 1\}$ and I_0 correspondingly.

$$\alpha^* = \frac{1}{m}|I_1|$$
$$\mu_0^* = \frac{1}{|I_0|}\sum_{j \in I_0} x^j, \quad \mu_1^* = \frac{1}{|I_1|}\sum_{j \in I_1} x^j$$
$$V^* = \frac{1}{m}\sum_{j=1}^{m} (x^j - \mu_{y^j}) \otimes (x^j - \mu_{y^j})$$



3. Maximum Likelihood estimator, consistency

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

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

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

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

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

Consider $L(\theta, \mathcal{T}^m) = L(\theta) + \big[L(\theta, \mathcal{T}^m) - L(\theta)\big]$

- The model should be identifiable, i.e. $\theta_0 = \arg\max_{\theta \in \Theta} L(\theta)$
- Ensure that he Uniform Law of Large Numbers (ULLN) holds, i.e.

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

  for any $\epsilon > 0$.

3. Maximum Likelihood estimator, consistency

Identifiability of the model θ_0 is easy to prove if $p_{\theta_0}(x) \not\equiv p_\theta(z)$ holds $\forall \theta \neq \theta_0$.

$$L(\theta_0) - L(\theta) = D_{KL}(p_{\theta_0}(x) \parallel p_\theta(x)) \geqslant 0$$

and becomes zero if and only if $\theta = \theta_0$.

ULLN can be ensured e.g. by requiring that

- $L(\theta, \mathcal{T}^m)$ is continuous in θ and $\Theta \subset \mathbb{R}^k$ is compact.
- $L(\theta, \mathcal{T}^m)$ can be upper bounded: $\log p_\theta(x) \leqslant d(x)$ $\forall \theta$ with $\mathbb{E}_{\theta_0} d(x) < \infty$.

4. The Expectation Maximisation Algorithm

Unsupervised generative learning:

- The joint p.d. $p_\theta(x, y)$, $\theta \in \Theta$ is known up to the parameter $\theta \in \Theta$,
- given training data $\mathcal{T}^m = \{x^j \in \mathcal{X} \mid i = 1, 2, \ldots, m\}$ i.i.d. generated from p_{θ_0}.

How shall we implement the MLE

$$\theta_m(\mathcal{T}^m) = \arg\max_{\theta \in \Theta} \frac{1}{m} \sum_{x \in \mathcal{T}^m} \log p_\theta(x) = \arg\max_{\theta \in \Theta} \mathbb{E}_{\mathcal{T}^m}\Big[\log \sum_{y \in \mathcal{Y}} p_\theta(x, y)\Big]$$

- If θ is a single parameter or a vector of homogeneous parameters $\Rightarrow$ maximise the log-likelihood directly.
- If θ is a collection of heterogeneous parameters $\Rightarrow$ apply the **Expectation Maximisation Algorithm** (Schlesinger, 1968, Sundberg, 1974, Dempster, Laird, and Rubin, 1977)

4. The Expectation Maximisation Algorithm

EM approach:

- Introduce auxiliary variables $\alpha_x(y) \geqslant 0$, for each $x \in \mathcal{T}^m$, s.t. $\sum_{y \in \mathcal{Y}} \alpha_x(y) = 1$
- Construct a lower bound of the log-likelihood $L(\theta, \mathcal{T}^m) \geqslant L_B(\theta, \alpha, \mathcal{T}^m)$
- Maximise this lower bound by block-wise coordinate ascent.

Construct the bound:

$$L(\theta, \mathcal{T}^m) = \mathbb{E}_{\mathcal{T}^m}\Big[\log \sum_{y \in \mathcal{Y}} p_\theta(x,y)\Big] = \mathbb{E}_{\mathcal{T}^m}\Big[\log \sum_{y \in \mathcal{Y}} \frac{\alpha_x(y)}{\alpha_x(y)} p_\theta(x,y)\Big] \geqslant$$

$$L_B(\theta, \alpha, \mathcal{T}^m) = \mathbb{E}_{\mathcal{T}^m} \sum_{y \in \mathcal{Y}} \Big[\alpha_x(y) \log p_\theta(x,y) - \alpha_x(y) \log \alpha_x(y)\Big]$$

The following equivalent representation shows the difference between $L(\theta, \mathcal{T}^m)$ and $L_B(\theta, \alpha, \mathcal{T}^m)$:

$$L_B(\theta, \alpha, \mathcal{T}^m) = \mathbb{E}_{\mathcal{T}^m}\big[\log p_\theta(x)\big] - \mathbb{E}_{\mathcal{T}^m}\big[D_{KL}(\alpha_x(y) \,\|\, p_\theta(y \,|\, x))\big]$$

4. The Expectation Maximisation Algorithm

Maximise $L_B(\theta, \alpha, \mathcal{T}^m)$ by block-coordinate ascent:

Start with some $\theta^{(0)}$ and iterate

E-step Fix the current $\theta^{(t)}$, maximise $L_B(\theta^{(t)}, \alpha, \mathcal{T}^m)$ w.r.t. α-s. This gives

$$\alpha_x^{(t)}(y) = p_{\theta^{(t)}}(y \mid x).$$

M-step Fix the current $\alpha^{(t)}$ and maximise $L_B(\theta, \alpha^{(t)}, \mathcal{T}^m)$ w.r.t. θ.

$$\theta^{(t+1)} = \arg\max_{\theta \in \Theta} \mathbb{E}_{\mathcal{T}^m}\Big[\sum_{y \in \mathcal{Y}} \alpha_x^{(t)}(y) \log p_\theta(x, y)\Big]$$

This is equivalent to solving the MLE for annotated training data.

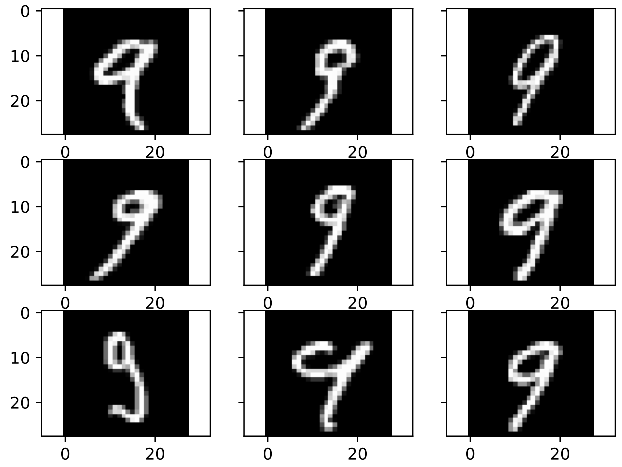
Claims:

- The bound is tight if $\alpha_x(y) = p_\theta(y \mid x)$,
- The sequence of likelihood values $L(\theta^{(t)}, \mathcal{T}^m)$, $t = 1, 2, \ldots$ is increasing, and the sequence $\alpha^{(t)}$, $t = 1, 2, \ldots$ is convergent (under mild assumptions).

4. The Expectation Maximisation Algorithm

Example: Latent mode model (mixture) for images of digits

- $x = \{x_i \mid i \in D\}$ image on the pixel domain $D \in \mathbb{Z}^2$,
- $x_i \in \{0, 1, 2, \ldots, 255\}$
- $k \in K$ latent variable (mode indicator),
- joint distribution - Naive Bayes model



$$p(x, k) = p(k) \prod_{i \in D} p(x_i \mid k)$$

Learning problem: Given i.i.d. training data $\mathcal{T}^m = \{x^j \mid j = 1, 2, \ldots, m\}$, estimate the mode probabilities $p(k)$ and the conditional probabilities $p(x_i \mid k)$, $\forall x_i \in \mathcal{B}$, $k \in K$ and $i \in D$.

4. The Expectation Maximisation Algorithm

Applying the EM algorithm: Start with some model $p^{(0)}(k)$, $p^{(0)}(x_i \mid k)$ and iterate the following steps until convergence.

E-step Given the current model estimate $p^{(t)}(k)$, $p^{(t)}(x_i \mid k)$, compute the posterior mode probabilities for each image x in the training data $\mathcal{T}^m$

$$\alpha_x^{(t)}(k) = p^{(t)}(k \mid x) = \frac{p^{(t)}(k) \prod_{i \in D} p^{(t)}(x_i \mid k)}{\sum_{k'} p^{(t)}(k') \prod_{i \in D} p^{(t)}(x_i \mid k')}.$$

M-step Re-estimate the model by solving

$$\mathbb{E}_{\mathcal{T}^m}\Big[\sum_{k \in K} \alpha_x^{(t)}(k) \big[\log p(k) + \sum_{i \in D} \log p(x_i \mid k)\big]\Big] \to \max_p$$

This gives

$$p^{(t+1)}(k) = \mathbb{E}_{\mathcal{T}^m}\big[\alpha_x^{(t)}(k)\big]$$

$$p^{(t+1)}(x_i = b \mid k) = \frac{\mathbb{E}_{\mathcal{T}^m}\big[\alpha_x^{(t)}(k) \bigm| x_i = b\big]}{\mathbb{E}_{\mathcal{T}^m}\big[\alpha_x^{(t)}(k)\big]}$$

4. The Expectation Maximisation Algorithm

Additional reading:

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

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