Statistical Machine Learning (BE4M33SSU)
Unsupervised Learning

Czech Technical University in Prague
B. Flach, V. Franc

- Unsupervised learning
- Expectation Maximization algorithm

Unsupervised learning

Learning from unlabeled data $x_1, x_2, \ldots, x_m$

Typical goals

- Discover hidden structure (clusters, groupings)
- Reduce dimensionality
- Estimate the data density $p(x)$
- Detect anomalies and outliers
- Learn useful representations / features

Typical applications

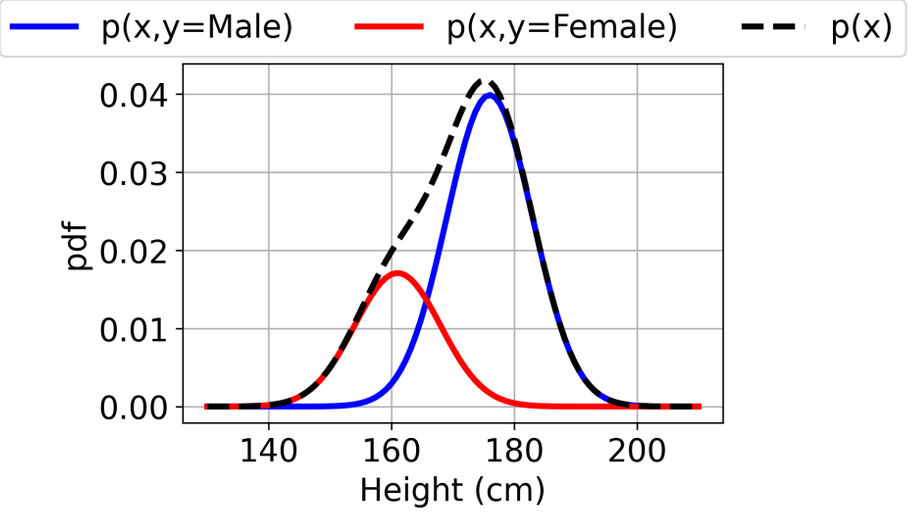
- Customer segmentation
- Recommendation systems
- Fraud and anomaly detection
- Topic modeling
- Bioinformatics (gene expression)
- Latent signal reconstruction (PET)

Typical methods: k-means, hierarchical clustering, DBSCAN, PCA, t-SNE, autoencoders, ICA, Expectation Maximization algorithm.

Unsupervised learning

Example 1 (Supervised learning). *Consider a joint distribution over $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} = \mathbb{R}$ and $\mathcal{Y} = \{1, \ldots, Y\}$:*

$$p(x, y; \theta) = p(y) \frac{1}{\sqrt{2\pi}\sigma_y} \exp\left(-\frac{(x - \mu_y)^2}{2\sigma_y^2}\right) \quad (1)$$



parametrized θ, which collects the mean and standard deviation (μ_y, σ_y) of each Gaussian component $y \in \mathcal{Y}$ together with the class probabilities $p(y)$.

The goal is to learn parameters θ from the training data $T_m = ((x_i, y_i) \in \mathcal{X} \times \mathcal{Y} \mid i = 1, \ldots, m)$ drawn i.i.d. from $p(x, y; \theta)$.

Example 2 (Unsupervised learning). *The goal is to find the parameters θ using the training data $T_m = (x_i \in \mathcal{X} \mid i = 1, \ldots, m)$ i.i.d. generated from a Gaussian Mixture Model*

$$p(x; \theta) = \sum_{y \in \mathcal{Y}} p(x, y; \theta)$$

where $p(x, y; \theta)$ is given by (1). The labels y_i are not *observed.*

Unsupervised learning

Example 3 (Generating handwritten digits)**.**

Suppose we observe images of handwritten digits

$$x_1, x_2, \ldots, x_m,$$

each associated with a hidden variable z_i *that encodes shape and writing style. The data is assumed to be generated by a* ***latent variable model****:*

$$z_i \sim p(z) \quad \text{and} \quad x_i \sim p(x \,|\, z_i; \theta),$$

so that the marginal distribution of an image is

$$p(x; \theta) = \int p(z)\, p(x \,|\, z; \theta)\, dz.$$

We fix a simple prior $p(z)$ *on the latent space, e.g.* $\mathcal{N}(\mathbf{0}, \mathbb{I})$*, and a parametric likelihood* $p(x \,|\, z; \theta)$*, e.g.* $\mathcal{N}\big(\mu(z; \theta), \sigma^2 \mathbb{I}\big)$*, where* $\mu(\,\cdot\,; \theta) \colon \mathbb{R}^n \to \mathbb{R}^{h \times w}$ *is a parametrised mapping (e.g. a neural network).*

Can we estimate θ ***without ever observing the latent states*** z_i***?***

Supervised Generative Learning

Problem formulation. Given a parametric family of joint distributions $\{p(x,y;\theta) \mid \theta \in \Theta\}$ and a training set

$$T_m = \big((x_i, y_i) \in \mathcal{X} \times \mathcal{Y} \,\big|\, i = 1, \ldots, m\big),$$

we want to estimate the parameter θ by the *maximum likelihood estimator* (MLE):

$$e_{\mathrm{ML}}(T_m) \;=\; \arg\max_{\theta\in\Theta} L(\theta) \;=\; \arg\max_{\theta\in\Theta} \frac{1}{m} \sum_{i=1}^{m} \log p(x_i, y_i; \theta)$$

One-hot-label encoding The training label $y_i \in \mathcal{Y}$ can be encoded by a vector $\alpha_i = [\alpha_i(1), \alpha_i(2), \ldots, \alpha_i(Y)]$ such that $\alpha_i(y) = 1$ and $\alpha_i(y') = 0$, $\forall y' \neq y$.

The MLE problem can be than written as

$$e_{\mathrm{ML}}(T_m) \;=\; \arg\max_{\theta\in\Theta} \frac{1}{m} \sum_{i=1}^{m} \sum_{y\in\mathcal{Y}} \alpha_i(y) \log p(x_i, y; \theta)$$

Supervised Generative Learning

Example 4 (Fitting Gaussians). *Given training data* $T_m = \big((x_i, y_i) \in \mathbb{R} \times \mathcal{Y} \,\big|\, i = 1, \ldots, m\big)$, *estimate the parameters* $p(y)$, μ_y, σ_y *of the Gaussian distribution:*

$$p(x, y; \theta) = p(y)p(x \mid y; \theta) = p(y)\frac{1}{\sqrt{2\pi}\sigma_y} e^{-\frac{(x-\mu_y)^2}{2\sigma_y^2}}$$

The MLE leads to

$$e_{\mathrm{ML}}(T_m; \theta) = \operatorname*{arg\,max}_{p(y),\mu_y,\sigma_y} \frac{1}{m} \sum_{i=1}^{m} \log p(x_i, y_i; \theta) = \operatorname*{arg\,max}_{p(y),\mu_y,\sigma_y} \frac{1}{m} \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \alpha_i(y) \log p(x_i, y; \theta)$$

which has a closed form solution:

$$\begin{aligned}
p(y) &= \frac{1}{m} \sum_{i=1}^{m} \alpha_i(y) \\
\mu_y &= \sum_{i=1}^{m} \alpha_i(y) x_i \Big/ \left(\sum_{i=1}^{m} \alpha_i(y) \right) \\
(\sigma_y)^2 &= \sum_{i=1}^{m} [\alpha_i(y) (x_i - \mu_y)^2] \Big/ \left(\sum_{i=1}^{m} \alpha_i(y) \right)
\end{aligned}$$

Unsupervised learning

Problem formulation. Given a parametric family of joint distributions $\{p(x,y;\theta) \mid \theta \in \Theta\}$ and a training set

$$T_m = \big(x_i \in \mathcal{X} \,\big|\, i = 1,\ldots,m\big),$$

we want to estimate the parameter θ by the *maximum likelihood estimator* (MLE):

$$e_{\mathrm{ML}}(T_m) \;=\; \arg\max_{\theta\in\Theta} L(\theta) \;=\; \arg\max_{\theta\in\Theta} \frac{1}{m}\sum_{i=1}^{m} \log p(x_i;\theta) \;=\; \arg\max_{\theta\in\Theta} \frac{1}{m}\sum_{i=1}^{m} \log \sum_{y\in\mathcal{Y}} p(x_i,y;\theta).$$

Remark: For simplicity we assume in this lecture that $\mathcal{Y} = \{1,\ldots,Y\}$ is finite.

- If θ is a single parameter or a vector of *homogeneous* parameters $\Rightarrow$ maximise the log-likelihood directly by gradient ascent (provided it is differentiable in θ).
- If θ is a collection of *heterogeneous* parameters $\Rightarrow$ apply the **Expectation–Maximisation (EM) algorithm** (Schlesinger, 1968; Sundberg, 1974; Dempster, Laird, and Rubin, 1977).
- The EM algorithm transforms the original unsupervised problem into a sequence of supervised MLE tasks, which are typically much easier to solve.

The Expectation Maximization Algorithm

Input. Unlabeled training set $T_m = \big(x_i \in \mathcal{X} \,\big|\, i = 1, \ldots, m\big)$.

Start with a suitably chosen $\theta^{(0)}$ and iterate the following steps until convergence

E-step Fix the current $\theta^{(t)}$ and compute

$$\alpha_i^{(t)}(y) = p(y \mid x_i; \theta^{(t)}) \qquad i = 1, \ldots, m\,.$$

M-step Fix the current $\alpha^{(t)} = (\alpha_1^{(t)}, \ldots, \alpha_m^{(t)})$, use them as "soft" labels and solve the MLE task.

$$\theta^{(t+1)} = \arg\max_{\theta \in \Theta} \frac{1}{m} \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \alpha_i^{(t)}(y) \log p(x_i, y; \theta)$$

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

Claims:

- The sequence of likelihood values $L(\theta^{(t)}) = \frac{1}{m} \sum_{i=1}^{m} \log p(x_i; \theta^{(t)})$, $t = 1, 2, \ldots$ is increasing.
- The sequence of $\alpha_i^{(t)}$, $t = 1, 2, \ldots$ is convergent.

There is **no guarantee** that the EM algorithm converges to a global maximum of the log-likelihood. This underlines the importance of a suitable initialization.

The Expectation Maximization Algorithm

Example 5 (Fitting mixture of Gaussians)**.** *Given training data* $T_m = \big(x_i \in \mathbb{R} \,\big|\, i = 1, \ldots, m\big)$*, estimate the parameters* $p(y)$*,* μ_y*,* σ_y *of the GMM:*

$$p(x;\theta) = \sum_{y\in\mathcal{Y}} p(y)p(x \mid y;\theta) = \sum_{y\in\mathcal{Y}} p(y)\frac{1}{\sqrt{2\pi}\sigma_y} e^{-\frac{(x-\mu_y)^2}{2\sigma_y^2}}$$

Start with a suitably chosen $\theta^{(0)} = (p^{(0)}(y), \mu_y^{(0)}, \sigma_y^{(0)})$ *and iterate until convergence:*

E-step *Fix the current* $\theta^{(t)}$ *and compute*

$$\alpha_i^{(t)}(y) = p(y \mid x_i;\theta^{(t)}) = \frac{p^{(t)}(y)}{\sqrt{2\pi}\sigma_y} e^{-(x_i-\mu_y^{(t)})^2/(2\sigma_y^{(t)})^2} \Big/ \left(\sum_{\bar{y}\in\mathcal{Y}} \frac{p^{(t)}(\bar{y})}{\sqrt{2\pi}\sigma_{\bar{y}}^{(t)}} e^{-(x_i-\mu_{\bar{y}}^{(t)})^2/(2\sigma_{\bar{y}}^{(t)})^2} \right)$$

M-step *Fix the current* $\alpha^{(t)}$*, use them as "soft" labels and solve the MLE task.*

$$(p^{(t+1)}(y), \mu_y^{(t+1)}, \sigma_y^{(t+1)}) = \operatorname*{arg\,max}_{p(y),\mu_y,\sigma_y} \frac{1}{m} \sum_{i=1}^{m} \sum_{y\in\mathcal{Y}} \alpha_i^{(t)}(y) \log p(x_i, y;\theta)$$

which has a closed form solution:

$$p^{(t+1)}(y) = \frac{1}{m}\sum_{i=1}^{m} \alpha_i^{(t)}(y)\,,\;\; \mu_y^{(t+1)} = \frac{\sum_{i=1}^{m} \alpha_i^{(t)}(y) x_i}{\sum_{i=1}^{m} \alpha_i^{(t)}(y)}\,,\;\; \left(\sigma_y^{(t+1)}\right)^2 = \frac{\sum_{i=1}^{m} [\alpha_i^{(t)}(y)\big(x_i - \mu_y^{(t+1)}\big)^2]}{\sum_{i=1}^{m} \alpha_i^{(t)}(y)}$$

The Expectation Maximization Algorithm

- Introduce auxiliary variables $\alpha_i(y) \geq 0$, for each $(x_1, \ldots, x_m)$, s.t. $\sum_{y \in \mathcal{Y}} \alpha_i(y) = 1$
- Construct a lower bound of the log-likelihood $L(\theta) \geq L_B(\theta, \alpha)$
- Maximize this lower bound by block-wise coordinate ascent.

Construct the bound using the Jensen's inequality:

$$L(\theta) = \frac{1}{m} \sum_{i=1}^{m} \log \sum_{y \in \mathcal{Y}} p(x_i, y; \theta) = \frac{1}{m} \sum_{i=1}^{m} \log \sum_{y \in \mathcal{Y}} \frac{\alpha_i(y)}{\alpha_i(y)} p(x_i, y; \theta) \geq$$

$$L_B(\theta, \alpha) = \frac{1}{m} \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \alpha_i(y) \log \left[\frac{p(x_i, y; \theta)}{\alpha_i(y)} \right] = \frac{1}{m} \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \Big[\alpha_i(y) \log p(x_i, y; \theta) - \alpha_i(y) \log \alpha_i(y) \Big]$$

The following equivalent representation shows the difference between $L(\theta)$ and $L_B(\theta, \alpha)$:

$$L_B(\theta, \alpha) = \frac{1}{m} \sum_{i=1}^{m} \log \sum_{y \in \mathcal{Y}} p(x_i, y; \theta) - \frac{1}{m} \sum_{i=1}^{m} D_{KL}(\alpha_i(y) \parallel p(y \,|\, x_i; \theta))$$

We see that the lower bound is tight if $\alpha_i(y) = p(y \mid x_i; \theta)$ holds $\forall i \in \{1, \ldots, m\}$ and $\forall y \in \mathcal{Y}$.

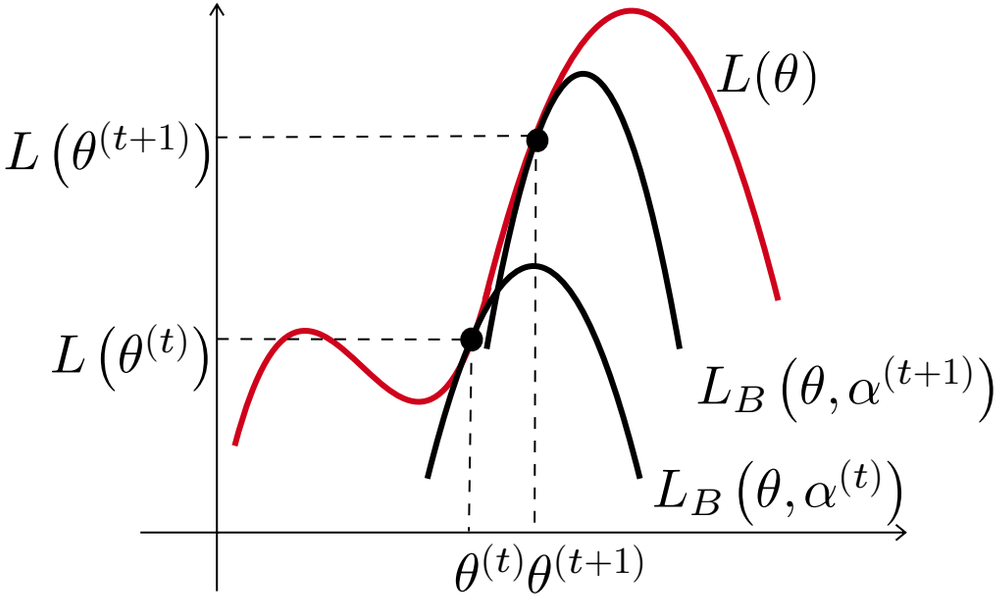
The Expectation Maximization Algorithm

- The EM implements the block-wise coordinate ascent of the lower bound

$$L(\theta) \geq L_B(\theta, \alpha) = \frac{1}{m} \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \Big[\alpha_i(y) \log p(x_i, y; \theta) - \alpha_i(y) \log \alpha_i(y) \Big], \qquad \forall \theta, \alpha$$

E-step Fix the current $\theta^{(t)}$ and compute

$$\alpha_i^{(t)}(y) = \underset{\substack{\alpha_i(y) \geq 0 \\ \sum_y \alpha_i(y) = 1}}{\arg\max} L_B(\theta^{(t)}, \alpha) = p(y \mid x_i; \theta^{(t)})$$



M-step Fix the current $\alpha^{(t)}$ and compute

$$\begin{aligned} \theta^{(t+1)} &= \underset{\theta \in \Theta}{\arg\max}\, L_B(\theta, \alpha^{(t)}) \\ &= \underset{\theta \in \Theta}{\arg\max}\, \frac{1}{m} \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \alpha_i^{(t)}(y) \log p(x_i, y; \theta) \end{aligned}$$

EM in engineering: three applications

The observed signal is a sum of contributions from unobserved sources – EM disentangles them.

Application	Observed signal	Latent variable	M-step
Wireless channel estimation (SAGE)	Antenna waveform: superposition of multipath echoes	Per-path complete waveform	Single-path matched filter over delay, Doppler, angle
PET reconstruction	Detector counts: sum of voxel emissions	Voxel of origin of each photon pair	Multiplicative Poisson-rate update per voxel
Multi-target tracking	Sensor returns: targets + clutter	Measurement-to-target association	Decoupled Kalman smoother per target

- **E-step everywhere:** attribute the observed evidence to candidate sources, weighted by current beliefs (interference cancellation = photon assignment = soft data association).
- **M-step everywhere:** re-estimate each source from its attributed evidence — a *supervised* sub-problem, easy to solve.
- **Why EM wins here:** the joint MLE is non-convex and high-dimensional; per-source updates are tractable.