MO433 - Unsupervised Learning
Introduction to Unsupervised Learning

Alexandre Xavier Falcão

Institute of Computing - UNICAMP

afalcao@ic.unicamp.br

Agenda

- What is unsupervised learning?
- The importance of the joint probability density function (pdf, distribution for simplicity).
- Overview of this course with emphasis on deep generative models.
- Basic concepts from Probability and Information Theory.

What is unsupervised learning?

Let $\mathrm{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ be a random vector, whose variables x_i, $i \in [1, n]$, are observations describing different measures of a phenomenon under study.

What is unsupervised learning?

Let $\mathrm{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ be a random vector, whose variables x_i, $i \in [1, n]$, are observations describing different measures of a phenomenon under study. Such variables might be:

- age, income, purchase frequency to discover customers with similar characteristics;
- expression levels of different genes to find co-regulated gene groups or genes that express together;
- latent features from input images to synthesize new images; etc.

What is unsupervised learning?

Let $\mathrm{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ be a random vector, whose variables x_i, $i \in [1, n]$, are observations describing different measures of a phenomenon under study. Such variables might be:

- age, income, purchase frequency to discover customers with similar characteristics;
- expression levels of different genes to find co-regulated gene groups or genes that express together;
- latent features from input images to synthesize new images; etc.

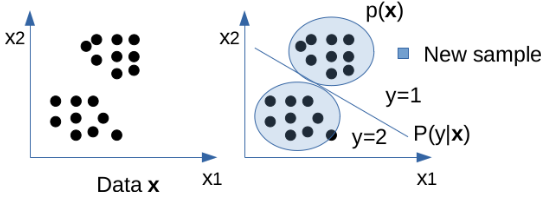
Unsupervised learning is the process of discovering the underlying structure (clusters, associations, latent factors) of a joint pdf $p(\mathrm{x})$ from N observed samples $\{\mathrm{x}^{(1)}, \mathrm{x}^{(2)}, \ldots, \mathrm{x}^{(N)}\}$, without access to labeled target variables.

The importance of $p(x)$

- Knowledge of $p(x)$ allows data analysis, synthesis, and efficient annotation by focusing human effort on more representative samples from high-density regions.

The importance of $p(\mathrm{x})$

- Knowledge of $p(\mathrm{x})$ allows data analysis, synthesis, and efficient annotation by focusing human effort on more representative samples from high-density regions.
- It enables better classification by utilizing the joint pdf $p(\mathrm{x}, y) = P(y \mid \mathrm{x})p(\mathrm{x})$ rather than only the conditional probability $P(y \mid \mathrm{x})$ represented by standard classifiers.

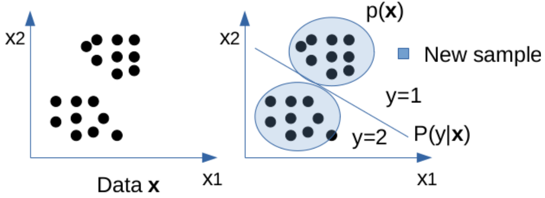


Samples in regions with low $p(\mathbf{x})$ could remain unclassified.

The importance of $p(\mathrm{x})$

- Knowledge of $p(\mathrm{x})$ allows data analysis, synthesis, and efficient annotation by focusing human effort on more representative samples from high-density regions.

- It enables better classification by utilizing the joint pdf $p(\mathrm{x}, y) = P(y \mid \mathrm{x})p(\mathrm{x})$ rather than only the conditional probability $P(y \mid \mathrm{x})$ represented by standard classifiers.



Samples in regions with low $p(\mathbf{x})$ could remain unclassified.

- This is particularly valuable for handling class imbalance, outlier detection, and uncertainty quantification.

What will you learn in this course?

- **Mathematical Foundations** — as needed to support core models and algorithms.
- **Dimensionality Reduction & Visualization** — uncover patterns and structure in high-dimensional data.
- **Clustering & Distribution Estimation** — learn methods to group data and model its probability structure.
- **Representation Learning** — with autoencoders, contrastive and non-contrastive self-supervised methods.
- **Deep Generative Models** — create realistic images with deep neural architectures.

Main Types of Deep Generative Models

Deep generative models aim to estimate the underlying pdf $p(\mathrm{x})$ to generate new, realistic samples.

- **Autoregressive Models**
  Factorize the joint pdf as a sequence of conditional distributions: $p(\mathrm{x}) = \prod_{i=1}^{n} p(x_i \mid x_{<i})$.
  Examples: PixelCNN, GPT.

- **Latent Variable Models**
  Introduce hidden variables z and marginalize them out:
  $p(\mathrm{x}) = \int_{-\infty}^{+\infty} p(\mathrm{x} \mid \mathrm{z}) p(\mathrm{z})\, dz$
  Examples: VAEs, GANs, stable diffusion.

- **Flow-based Models**
  Use invertible bijective vector-valued functions f between latent space and data: $\mathrm{x} = \mathrm{f}(\mathrm{z})$, so $p(\mathrm{x}) = p(\mathrm{z}) \left| \det \left(\frac{\partial \mathrm{f}^{-1}}{\partial \mathrm{x}} \right) \right|$.
  Examples: RealNVP, Glow.

Other Generative Modeling Approaches

Alternative methods use implicit or non-likelihood-based estimation techniques to model $p(\mathrm{x})$:

- **Energy-Based Models**
  Assign an energy score $\mathcal{E}(\mathrm{x})$ and define distribution as:
  $p(\mathrm{x}) = \frac{1}{Z}\exp(-\mathcal{E}(\mathrm{x}))$, with $Z = \sum_{\mathrm{x}} \exp\{-\mathcal{E}(\mathrm{x})\}$.
  Examples: Deep Boltzmann machines.

- **Score-Based Models**
  Estimate the gradient of log-density $\nabla_{\mathrm{x}} \log p(\mathrm{x})$, and use it in Langevin dynamics or reverse-time stochastic differential equation to sample data.
  Examples: Score-based diffusion models, noise conditional score network.

Each model type balances tractability, generation speed, training stability, and sample realism.

More Details About This Course

The course is under preparation and all essential course resources will be available online, including:

- Lecture slides.
- Student evaluation criteria.
- Downloadable datasets.
- Recommended bibliography.
- Additional reference materials and updates

Visit: `www.ic.unicamp.br/~afalcao/mo433`

Random variables

Let x be a continuous random variable, then its mean $E[x]$ and variance $\mathrm{Var}[x]$ are defined by

$$\begin{aligned} E[x] &= \mu = \int_{-\infty}^{+\infty} x p(x)\, dx \\ \mathrm{Var}[x] &= E[(x-\mu)^2] = \sigma^2 = \int_{-\infty}^{+\infty} (x-\mu)^2 p(x)\, dx \end{aligned}$$

where $p(x)$ is the pdf of x.

Let $\bar{x} = \frac{x_1 + x_2 + \ldots + x_n}{n}$ be the sample mean of n independent random variables.

Sample mean properties:

$$\begin{aligned} E[\bar{x}] &= \frac{E[x_1] + E[x_2] + \ldots + E[x_n]}{n} \\ \mathrm{Var}[\bar{x}] &= \frac{\mathrm{Var}[x_1] + \mathrm{Var}[x_2] + \ldots + \mathrm{Var}[x_n]}{n^2} \end{aligned}$$

Central Limit Theorem

Classical CLT (i.i.d. case): If $x_1, x_2, \ldots, x_n$ are independent and identically distributed with $E[x_i] = \mu$ and $\text{Var}[x_i] = \sigma^2$, then:

$$\frac{\bar{x} - \mu}{\sigma/\sqrt{n}} \xrightarrow{d} N(0,1) \text{ as } n \to \infty$$

where $\bar{x} = \frac{x_1 + x_2 + \ldots + x_n}{n}$ and:

$$\begin{aligned} E[\bar{x}] &= \mu \\ \text{Var}[\bar{x}] &= \frac{\sigma^2}{n} \end{aligned}$$

Generalized CLT: Even if x_i have different distributions, under certain regularity conditions (e.g., Lindeberg condition: no single variance dominates), the standardized sum still converges to a normal distribution – i.e., $\bar{x} \sim N(\mu, \frac{\sigma^2}{n})$ – with pdf:

$$p(\bar{x}) = \frac{\sqrt{n}}{\sqrt{2\pi}\sigma} \exp\left(-\frac{n(\bar{x} - \mu)^2}{2\sigma^2}\right)$$

Statistical dependency

For two random variables, x_1 and x_2, the joint pdf:

$$p(x_1, x_2) \quad = \quad p(x_1 \mid x_2)p(x_2) = p(x_2 \mid x_1)p(x_1)$$

Marginalization:

$$\begin{aligned} p(x_1) &= \int_{-\infty}^{+\infty} p(x_1, x_2)\, dx_2 = \int_{-\infty}^{+\infty} p(x_1 \mid x_2)p(x_2)\, dx_2 \\ p(x_2) &= \int_{-\infty}^{+\infty} p(x_1, x_2)\, dx_1 = \int_{-\infty}^{+\infty} p(x_2 \mid x_1)p(x_1)\, dx_1 \end{aligned}$$

Independence: x_1 and x_2 are independent if and only if:

$$p(x_1, x_2) \quad = \quad p(x_1)p(x_2); p(x_1 \mid x_2) = p(x_1); \text{and } p(x_2 \mid x_1) = p(x_2)$$

Dependence: If x_1 and x_2 are dependent, then knowing one variable provides information about the other.

Statistical dependency

Covariance:

$$\begin{aligned}\mathrm{Cov}(x_1, x_2) &= E[(x_1 - \mu_1)(x_2 - \mu_2)] \\ &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} (x_1 - \mu_1)(x_2 - \mu_2) p(x_1, x_2)\, dx_1\, dx_2\end{aligned}$$

where $\mathrm{Cov}(x_1, x_2) = 0$ when x_1 and x_2 are independent (uncorrelated).

Correlation coefficient:

$$\rho_{x_1 x_2} = \frac{\mathrm{Cov}(x_1, x_2)}{\sigma_{x_1}\sigma_{x_2}}$$

where $|\rho_{x_1 x_2}| \leq 1$ and $|\mathrm{Cov}(x_1, x_2)| \leq \sigma_{x_1}\sigma_{x_2}$ (Cauchy-Schwarz Inequality).

Entropy

Entropy measures uncertainty (how hard it is to predict) in **bits** (number of yes/no questions needed to guess the outcome).

$$
\begin{aligned}
H(x_1) &= -\int_{-\infty}^{+\infty} p(x_1) \log p(x_1)\, dx_1 \\
H(x_2) &= -\int_{-\infty}^{+\infty} p(x_2) \log p(x_2)\, dx_2 \\
H(x_1, x_2) &= -\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p(x_1, x_2) \log p(x_1, x_2)\, dx_1\, dx_2 \\
H(x_2 \mid x_1) &= -\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p(x_1, x_2) \log p(x_2 \mid x_1)\, dx_1\, dx_2 \\
H(x_1 \mid x_2) &= -\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p(x_1, x_2) \log p(x_1 \mid x_2)\, dx_1\, dx_2
\end{aligned}
$$

If x_1 and x_2 are independent, a neural network cannot predict one given the other: $H(x_2 \mid x_1) = H(x_2)$ and $H(x_1 \mid x_2) = H(x_1)$.

Kullback-Leibler (KL) Divergence

The KL divergence measures how much one probability distribution diverges from the other.

$$
\begin{aligned}
D_{KL}(p(x_1)||p(x_2)) &= \int_{-\infty}^{+\infty} p(x_1) \log \frac{p(x_1)}{p(x_2)}\, dx_1 \\
&= \int_{-\infty}^{+\infty} p(x_1) \log p(x_1)\, dx_1 - \\
&\quad \int_{-\infty}^{+\infty} p(x_1) \log p(x_2)\, dx_1 \\
&= -H(x_1) - E_{x_1}[\log p(x_2)].
\end{aligned}
$$

Properties:

- $D_{KL}(p||q) \geq 0$ with equality iff $p = q$ (Gibbs' inequality).
- $D_{KL}(p||q) \neq D_{KL}(q||p)$ (asymmetric).
- Measures information lost when approximating p with q.

Mutual Information

Mutual information measures the amount of information shared between x_1 and x_2 (how much knowing one reduces uncertainty about the other).

$$
\begin{aligned}
I(x_1; x_2) &= H(x_1) + H(x_2) - H(x_1, x_2) \\
&= H(x_1) - H(x_1 \mid x_2) \\
&= H(x_2) - H(x_2 \mid x_1) \\
&= D_{KL}(p(x_1, x_2) || p(x_1) p(x_2)) \\
&= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p(x_1, x_2) \log \frac{p(x_1, x_2)}{p(x_1) p(x_2)} \, dx_1 \, dx_2
\end{aligned}
$$

If x_1 and x_2 are independent, a neural network cannot predict one given the other: $I(x_1; x_2) = 0$.

A practical example

Let x_1 be **interest rates** and x_2 be **housing prices**. One can design a neural network $f_\theta(x_1) \approx E[x_2 \mid x_1]$ given that:

$$\begin{aligned}
I(x_1; x_2) &> 0 &\Rightarrow&\quad x_1 \text{ and } x_2 \text{ share information} \\
H(x_2 \mid x_1) &< H(x_2) &\Rightarrow&\quad x_1 \text{ reduces uncertainty about } x_2 \\
E[x_2 \mid x_1] &\neq E[x_2] &\Rightarrow&\quad \text{conditional expectation varies with } x_1
\end{aligned}$$

A practical example

Let x_1 be **interest rates** and x_2 be **housing prices**. One can design a neural network $f_\theta(x_1) \approx E[x_2 \mid x_1]$ given that:

$$
\begin{aligned}
I(x_1; x_2) &> 0 &\Rightarrow& \quad x_1 \text{ and } x_2 \text{ share information} \\
H(x_2 \mid x_1) &< H(x_2) &\Rightarrow& \quad x_1 \text{ reduces uncertainty about } x_2 \\
E[x_2 \mid x_1] &\neq E[x_2] &\Rightarrow& \quad \text{conditional expectation varies with } x_1
\end{aligned}
$$

This neural network learns the conditional expectation function by minimizing the Mean Squared Error:

$$
\begin{aligned}
\text{MSE} &= \frac{1}{N} \sum_{i=1}^{N} (f_\theta(x_{1,i}) - x_{2,i})^2 \\
\arg\min_{f_\theta} E[(f_\theta(x_1) - x_2)^2] &= E[x_2 \mid x_1].
\end{aligned}
$$

Exercise: Verify codes 1-4 of this lecture and play with the neural network.