Supervised vs. Unsupervised Learning

CSE 4334 / 5334 Data Mining
Spring 2019

Won Hwa Kim

Part of the contents borrowed from Prof. Mark Craven / Prof. David Page Jr. at UW-Madison

Unsupervised Learning

Representation of Objects in Machine Learning

- **Given $\mathbf{x} = (x_1, x_2, \cdots x_N)$, $x_i \in \mathbb{R}^k$ find patterns in the data**

- An instance x (a specific object) represented by k dimensional features
- Each x_i is a coordinate in the feature space (Feature Representation)

Examples…

- Text document: frequency of words (i.e., bag of words)
- Images: color histogram
- Medical information: medical test results

Unsupervised Learning

Training

- **Given $\mathbf{x} = (x_1, x_2, \cdots x_N)$, $x_i \in \mathbb{R}^k$ find patterns in the data**

- Training data is a set of instances for learning (training) phase
- Usually assumed that the data are sampled from an unknown distribution
- **Independent and Identically Distributed (i.i.d)**
- Learning from the past (experience)

Testing

- Inference, estimation, classification
- Predict future based on the past

Unsupervised Learning

Unsupervised…

- We have data only and no labels.
- What can we learn / infer from the data?
- We can learn what the data looks like, e.g., shape and distribution.

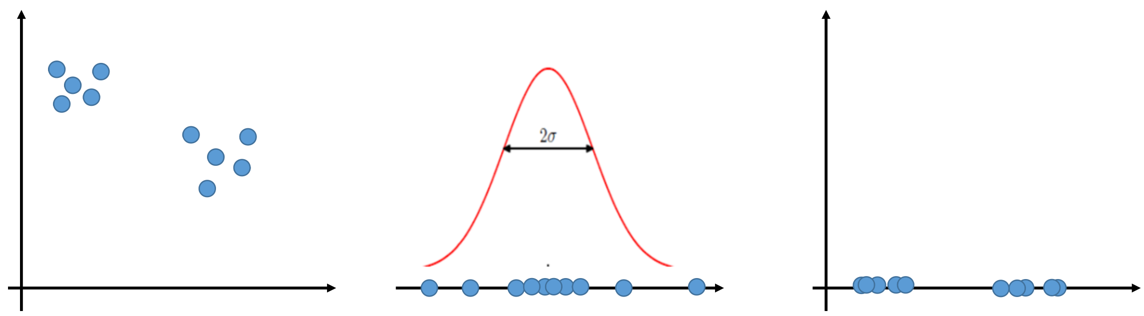
Let's try to…

- Model the probability distribution given finite set of observations
- Density estimation, clustering
- Dimension reduction, anomaly detection

Unsupervised Learning

Unsupervised…

- Given $\mathbf{x} = (x_1, x_2, \cdots x_N)$, $x_i \in \mathbb{R}^k$ find patterns in the data

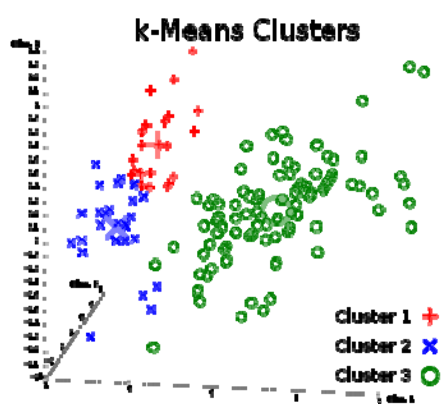


Unsupervised Learning

K-means

Given: i.i.d. $\mathbf{x} = (x_1, x_2, \cdots x_N)$, $x_i \in \mathbb{R}^d$ and a parameter k

- Partition N observations into k clusters in which each observation belongs to the cluster with the nearest mean
- NP-hard problem but heuristics exist
- This is not k-nearest neighbor classification



Unsupervised Learning

K-means

Given: i.i.d. $\mathbf{x} = (x_1, x_2, \cdots x_N)$, $x_i \in \mathbb{R}^d$ **and a parameter** k

Step 1: Select k cluster centers, c_1, c_2, … c_k

Step 2: Assignment step: for each point in x, determine its cluster based on the distance to the centers

Step 3: Update step: update all cluster centers as the centers of their clusters

$$c_i^{t+1} = \frac{1}{|S_i^t|} \sum_{x_j \in S_i^t} x_j$$

Step 4: Repeat 2 and 3 until converges

Unsupervised Learning

K-means

- Will it converge?

Yes

- Will it find the global optimal?

Not guaranteed

- How to choose initial centers?

make sure that they are far apart…

- What k shall we use?

domain knowledge / k that minimizes the error

Unsupervised Learning

Likelihood function

- a function of parameters of a statistical model given data

Given: *independent and identically distributed* (i.i.d.) $\mathbf{x} = (x_1, x_2, \cdots x_N)$, $x_i \in \mathbb{R}^k$

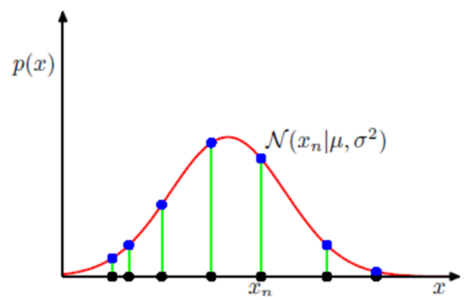
$$L(\theta|\mathbf{x}) = f(x|\theta)$$

$$\hat{\theta}(\mathbf{x}) = \text{argmax}_\theta L(\theta|\mathbf{x})$$

Probability distribution of a dataset

$$N(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{1}{2\sigma^2}(x-\mu)^2)$$

$$L(\mu,\sigma) = p(\mathbf{x}|\mu,\sigma^2) = \prod_{i=1}^{N} N(x_i|\mu,\sigma^2)$$



Unsupervised Learning

Log-likelihood of a Gaussian distribution

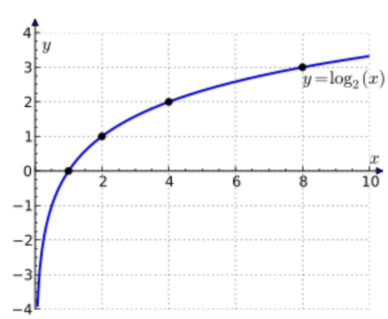
Given: *independent and identically distributed* (i.i.d.) $\mathbf{x} = (x_1, x_2, \cdots x_N)$, $x_i \in \mathbb{R}^k$

$$L(\mu,\sigma) = p(\mathbf{x}|\mu,\sigma^2) = \prod_{i=1}^{N} N(x_i|\mu,\sigma^2)$$

$$\ln p(\mathbf{x}|\mu,\sigma^2) = \ln \prod_{i=1}^{N} N(x_i|\mu,\sigma^2)$$

$$= \sum_{i=1}^{N} \ln N(x_i|\mu,\sigma^2)$$

$$= -\frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i-\mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi)$$



Unsupervised Learning

Maximum (Log) Likelihood Estimation

$$\ell(\mu, \sigma) = \ln p(\mathbf{x}|\mu, \sigma^2) = -\frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i - \mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi)$$

$$\mu_{MLE}, \sigma_{MLE} = \text{argmax}_{\mu,\sigma} \ln p(\mathbf{x}|\mu, \sigma^2)$$

$$\mu_{MLE} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\sigma^2_{MLE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_{MLE})^2$$

Unsupervised Learning

Binary Variable (Bernoulli trials)

- Single binary random variable $x \in \{0, 1\}$
- E.g., flipping a coin
- The probability of $x = 1$ denoted by paramter μ

$$p(x = 1|\mu) = \mu, \qquad 0 \le \mu \le 1$$
$$p(x = 0|\mu) = 1 - \mu$$

- The probability distribution over x (Bernoulli distribution) written as

$$Bern(x|\mu) = \mu^x(1-\mu)^{1-x}$$
$$\mathbb{E}[x] = \mu, var[x] = \mu(1-\mu)$$

Unsupervised Learning

Binary Variable (Bernoulli trials)

- Single binary random variable $x \in \{0,1\}$
- E.g., flipping a coin
- The probability of $x = 1$ denoted by paramter p

$$
\begin{aligned}
L(p|x) &= p^{x_1}(1-p)^{1-x_1} \cdots p^{x_N}(1-p)^{1-x_N} \\
&= p^{x_1}p^{x_2} \cdots p^{x_N}(1-p)^{1-x_1}(1-p)^{1-x_2} \cdots (1-p)^{1-x_N} \\
&= p^{(x_1+x_2 \cdots x_N)}(1-p)^{(N-x_1-x_2 \cdots -x_N)}
\end{aligned}
$$

Likelihood function: a function of the **parameters** of a statistical model given **data**.

Unsupervised Learning

Binary Variable (Bernoulli trials)

- Single binary random variable $x \in \{0,1\}$
- E.g., flipping a coin
- The probability of $x = 1$ denoted by paramter p

$$
\begin{aligned}
\ell(p|x) = \ln L(p|x) &= \ln p(\sum_{i=1}^{N} x_i) + \ln(1-p)(N - \sum_{i=1}^{N} x_i) \\
&= N(\bar{x}\ln p + (1-\bar{x})\ln(1-p))
\end{aligned}
$$

$$
\frac{\partial}{\partial p}\ell(p|x) = N\frac{\bar{x}-p}{p(1-p)}
$$

$$
p^* = \bar{x}
$$

Unsupervised Learning

Histogram (non-parametric)

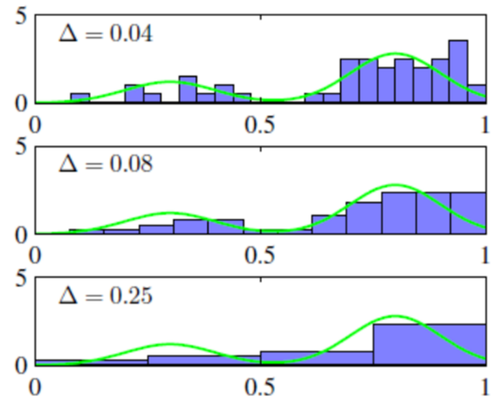
- Divide the domain into multiple bins
- Probability of a sample falling into each bin

$$p_i = \frac{n_i}{N\Delta_i}$$

# of observations

normalization

Bin width



Unsupervised Learning

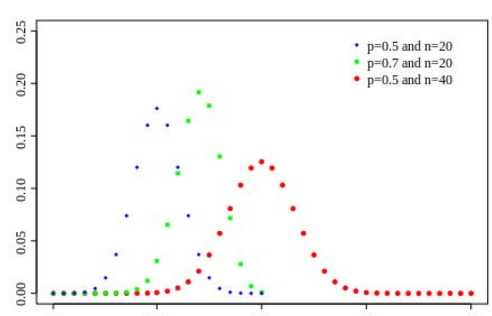
Binomial distribution

- discrete probability distribution of the number of successes in a sequence of n independent experiments

$$\text{Bin}(K|N,P) = \frac{N!}{K!(N-K)!}P^K(1-P)^{N-K}$$

$$\mathbb{E}[K] = NP$$
$$var[K] = NP(1-P)$$

K number of success out of N trials, with p chance of success



Unsupervised Learning

Prob. from histogram

$$p_i = \frac{n_i}{N\Delta_i}$$

Kernel Density Estimation (non-parametric)

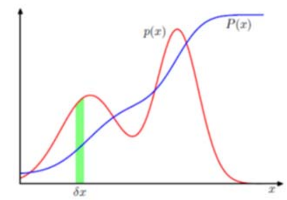
- P: probability of falling with in R
- Need to estimate p(x) given N data points in d-dimensional space

$$P = \int_R p(x)dx$$

K data points falling in R

$$\mathrm{Bin}(K|N,P) = \frac{N!}{K!(N-K)!}P^K(1-P)^{N-K}$$

$$\mathbb{E}[K/N] = P,\ var[K/N] = P(1-P)/N$$



$$K \simeq NP$$ ← For large N

Estimated Probability

$$P \simeq p(x)V$$ ← Sufficiently small R yielding constant prob in a unit volume V

$$\boxed{p(x) = \frac{K}{NV}}$$

Unsupervised Learning

Kernel Density Estimation (non-parametric)

- P: probability of falling with in R
- Need to estimate p(x) given N data points

$$\boxed{p(x) = \frac{K}{NV}}$$

$$k(u) = \begin{cases} 1 & |u_i| \leq 1/2, \quad i = 1, \cdots, d \\ 0 & o.w. \end{cases}$$

← Parzen window: a kernel function represented as a hypercube

$$K = \sum_{i=1}^{N} k\left(\frac{x - x_i}{h}\right)$$

← Total # of data points in the cube (**side = h**) centered at x

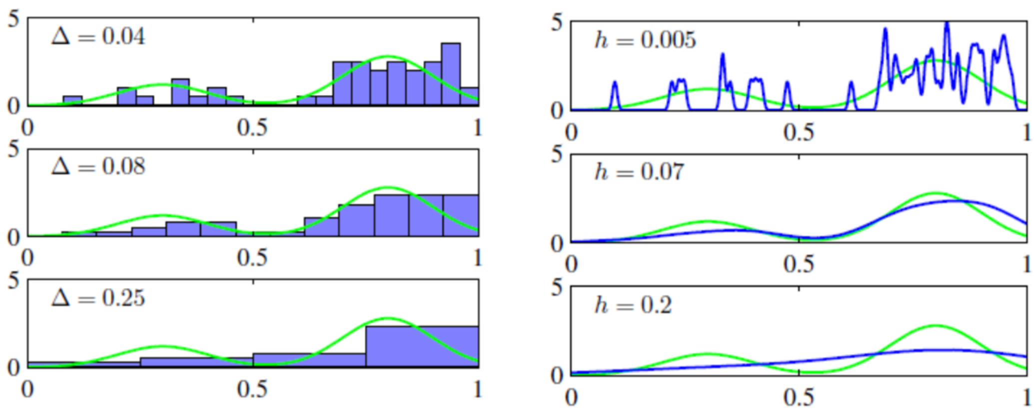
$$p(x) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{h^d}k\left(\frac{x - x_i}{h}\right)$$

← Estimated density at x

Unsupervised Learning

Kernel Density Estimation (non-parametric)

- P: probability of falling within R
- Need to estimate p(x) given N data points

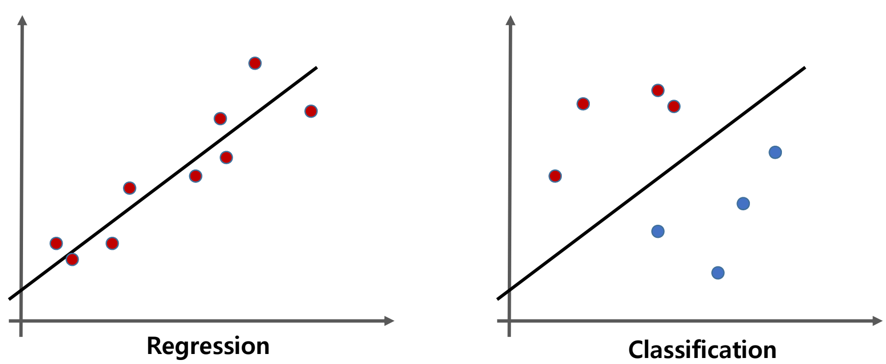


Supervised Learning

Dataset with Labels

- Given $\mathbf{x} = (x_1, x_2, \cdots x_N)$, $x_i \in \mathbb{R}^k$ and labels y_i, find patterns in the data

- Classification: label given as a class
- Regression: label given as a continuous variable



Supervised Learning

Training

- **Given $\mathbf{x} = (x_1, x_2, \cdots x_N)$, $x_i \in \mathbb{R}^k$ and labels y_i, find patterns in the data**

- Training data is a set of instances for learning (training) phase

- Usually assumed that the data are sampled from an unknown distribution

- Independent and Identically Distributed (i.i.d)

- Learning from the past (experience)

Testing

- Inference, estimation, classification

- Predict future based on the past

Supervised Learning

Supervised…

- We have data and labels.

- Experience with a teacher!

Let's try to…

- Make a hypothesis, and find a model that best fits the data

Given a training set of instances X, find a function that best maps X to labels y.

Supervised Learning

Methods for supervised learning

- Naïve Bayes classifier
- k-NN classifier
- Decision tree
- Artificial Neural Network
- Ensembles of classifiers

Supervised Learning

Naïve Bayes Classifier

- Conditional Probabilistic Model
- Assume that each feature is independent from each other
- Maximum Likelihood
- Requires relatively small training set

Supervised Learning

Naïve Bayes Classifier

- Given $\mathbf{x} = (x_1, x_2, \cdots x_N)$, $x_i \in \mathbb{R}^p$ and labels $y_i \in H$, find patterns in the data

$$p(h_k | x_1, \cdots, x_N)$$

Probability of class k given x

$$p(h_k|\mathbf{x}) = \frac{p(h_k)p(\mathbf{x}|h_k)}{p(\mathbf{x})}$$

Conditional probability decomposed using Bayes Theorem

$$posterior = \frac{prior \times likelihood}{evidence}$$

Evidence independent from y

$$\begin{aligned} y_{MAP} &= \text{argmax}_{h_k} p(h_k|\mathbf{x}) \\ &= \text{argmax}_{h_k} p(h_k)p(\mathbf{x}|h_k) \end{aligned}$$

Supervised Learning

Bayes Theorem
$p(A|B) = \frac{p(B|A)}{p(B)}$

Naïve Bayes Classifier

- Given $\mathbf{x} = (x_1, x_2, \cdots x_N)$, $x_i \in \mathbb{R}^p$ and labels $y_i \in H$, find patterns in the data

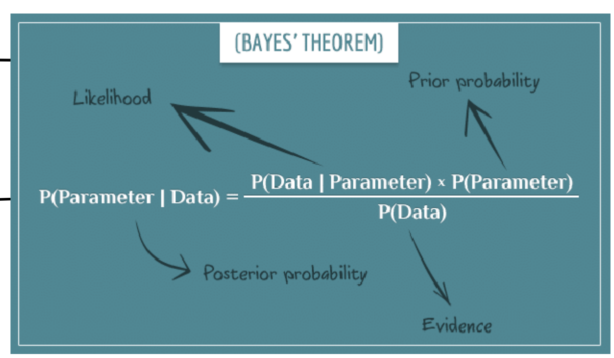
$$p(h_k | x_1, \cdots, x_N)$$

Probability of class k given x

$$p(h_k|\mathbf{x}) = \frac{p(h_k)p(\mathbf{x}|h_k)}{p(\mathbf{x})}$$

$$posterior = \frac{prior \times likelihood}{evidence}$$

$$\begin{aligned} y_{MAP} &= \text{argmax}_{h_k} p(h_k|\mathbf{x}) \\ &= \text{argmax}_{h_k} p(h_k)p(\mathbf{x}|h_k) \end{aligned}$$



Supervised Learning

Bayes Theorem
$p(A|B) = \frac{p(B|A)}{p(B)}$

Naïve Bayes Classifier

- Given $\mathbf{x} = (x_1, x_2, \cdots x_N)$, $x_i \in \mathbb{R}^p$ and labels $y_i \in H$, find patterns in the data

$$y_{MAP} = \text{argmax}_{h_k} p(h_k|\mathbf{x}) = \text{argmax}_{h_k} p(h_k)p(\mathbf{x}|h_k)$$

Maximum a posteriori (MAP)

$$y_{ML} = \text{argmax}_{h_k} p(\mathbf{x}|h_k)$$

With equally probable a priori, it becomes maximum likelihood (ML)

$$\begin{aligned} p(h_k, x_1, \cdots x_N) &= p(x_1, x_2, \cdots, x_n, h_k) \\ &= p(x_1|x_2, \cdots, x_N, h_k)p(x_2, \cdots, x_N, h_k) \\ &= p(x_1|x_2, \cdots, x_N, h_k)p(x_2|x_3, \cdots, x_N, h_k) \cdots \\ &\quad p(x_{N-1}|x_N, h_k)p(x_N|h_k)p(h_k) \end{aligned}$$

Using chain rule

$$P(X = x \mid Y = y) = \frac{P(X = x, Y = y)}{P(Y = y)}$$

Supervised Learning

Bayes Theorem
$p(A|B) = \frac{p(B|A)}{p(B)}$

Naïve Bayes Classifier

- Given $\mathbf{x} = (x_1, x_2, \cdots x_N)$, $x_i \in \mathbb{R}^p$ and labels $y_i \in H$, find patterns in the data

$$y_{MAP} = \text{argmax}_{h_k} p(h_k)p(\mathbf{x}|h_k)$$

Recall conditional distribution

$$P(X = x \mid Y = y) = \frac{P(X = x, Y = y)}{P(Y = y)}$$

$$p(h_k, x_1, \cdots x_N) = p(x_1|x_2, \cdots, x_N, h_k)p(x_2|x_3, \cdots, x_N, h_k) \cdots p(x_{N-1}|x_N, h_k)p(x_N|h_k)p(h_k)$$

$$p(x_i|x_{i+1}, \cdots, x_N, h_k) = p(x_i|h_k)$$

Because of independence

$$\begin{aligned} p(h_k|\mathbf{x}) &\propto p(h_k, x_1, x_2, \cdots x_N) \\ &\propto p(h_k)p(x_1|h_k)p(x_2|h_k) \cdots p(x_N|h_k) \\ &\propto p(h_k)\Pi_{i=1}^{N} p(x_i|h_k) \end{aligned}$$

Supervised Learning

Bayes Theorem
$p(A|B) = \frac{p(B|A)}{p(B)}$

Naïve Bayes Classifier

- Given $\mathbf{x} = (x_1, x_2, \cdots x_N)$, $x_i \in \mathbb{R}^p$ and labels $y_i \in H$, find patterns in the data

$$y_{MAP} = \text{argmax}_{h_k} p(h_k) p(\mathbf{x}|h_k)$$

$$\begin{aligned} p(h_k|\mathbf{x}) &\propto p(h_k, x_1, x_2, \cdots x_N) \\ &\propto p(h_k)p(x_1|h_k)p(x_2|h_k)\cdots p(x_N|h_k) \\ &\propto p(h_k)\Pi_{i=1}^N p(x_i|h_k) \end{aligned}$$

$$p(h_k|\mathbf{x}) = \frac{1}{Z} p(h_k)\Pi_{i=1}^N p(x_i|h_k)$$

Normalize it to be a probability

$$Z = \sum_k p(h_k) p(\mathbf{x}|h_k)$$

Supervised Learning

Bayes Theorem
$p(A|B) = \frac{p(B|A)}{p(B)}$

Naïve Bayes Classifier

- Given $\mathbf{x} = (x_1, x_2, \cdots x_N)$, $x_i \in \mathbb{R}^p$ and labels $y_i \in H$, find patterns in the data

$$y_{MAP} = \text{argmax}_{h_k} p(h_k) p(\mathbf{x}|h_k)$$

$$p(h_k|\mathbf{x}) = \frac{1}{Z} p(h_k)\Pi_{i=1}^N p(x_i|h_k) \qquad Z = \sum_k p(h_k) p(\mathbf{x}|h_k)$$

Inference / Prediction using Naïve Bayes

Maximum a posteriori (MAP)

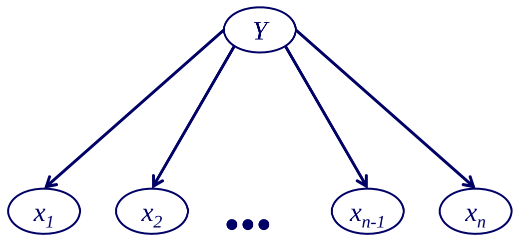
$$\hat{y} = \text{argmax}_k p(h_k)\Pi_{i=1}^k p(x_i|h_k)$$

$$\begin{aligned} y_{MAP} &= \text{argmax}_{h_k} p(h_k|\mathbf{x}) \\ &= \text{argmax}_{h_k} p(h_k) p(\mathbf{x}|h_k) \end{aligned}$$

Supervised Learning

Naïve Bayes Classifier

- One very simple BN approach for supervised tasks is *naïve Bayes*
- In naïve Bayes, we assume that all features X_i are conditionally independent given the class Y



$$\hat{y} = \operatorname{argmax}_k p(h_k) \Pi_{i=1}^{k} p(x_i | h_k)$$