Knowledge Engineering and Expert Systems

Lecture Notes on Machine Learning

Matteo Mattecci

matteucci@elet.polimi.it

Department of Electronics and Information

Politecnico di Milano

Supervised Learning

– Bayes Classifiers –

Density-Based Classifiers

You want to predict output Y which has arity n_Y and values $v_1, v_2, \ldots, v_{n_y}$.

- Assume there are m input attributes called $X_1, X2, \ldots, X_m$
- Break the dataset into n_Y smaller datasets called $DS_1, DS_2, \ldots, DS_{n_y}$
- Define $DS_i =$ Records in which$Y = v_i$
- For each DS_i learn the Density Estimator M_i to model the input distribution among the $Y = v_i$ records
  - M_i estimates $P(X_1, X_2, \ldots, X_m | Y = v_i)$

Density-Based Classifiers

You want to predict output Y which has arity n_Y and values $v_1, v_2, \ldots, v_{n_y}$.

- Assume there are m input attributes called $X_1, X2, \ldots, X_m$
- Break the dataset into n_Y smaller datasets called $DS_1, DS_2, \ldots, DS_{n_y}$
- Define $DS_i =$ Records in which$Y = v_i$
- For each DS_i learn the Density Estimator M_i to model the input distribution among the $Y = v_i$ records
  - M_i estimates $P(X_1, X_2, \ldots, X_m | Y = v_i)$

Idea 1:

When you get a new set of input values $(X_1 = u_1, X_2 = u_2, \ldots, X_m = u_m)$ predict the value of Y that makes $P(X_1, X_2, \ldots, X_m | Y = v_i)$ most likely

$$\hat{Y} = \arg\max_{v_i} P(X_1, X_2, \ldots, X_m | Y = v_i)$$

Density-Based Classifiers

You want to predict output Y which has arity n_Y and values $v_1, v_2, \ldots, v_{n_y}$.

- Assume there are m input attributes called $X_1, X2, \ldots, X_m$
- Break the dataset into n_Y smaller datasets called $DS_1, DS_2, \ldots, DS_{n_y}$
- Define $DS_i =$ Records in which$Y = v_i$
- For each DS_i learn the Density Estimator M_i to model the input distribution among the $Y = v_i$ records
  - M_i estimates $P(X_1, X_2, \ldots, X_m | Y = v_i)$

Idea 1:

When you get a new set of input values $(X_1 = u_1, X_2 = u_2, \ldots, X_m = u_m)$ predict the value of Y that makes $P(X_1, X_2, \ldots, X_m | Y = v_i)$ most likely

$$\hat{Y} = \arg\max_{v_i} P(X_1, X_2, \ldots, X_m | Y = v_i)$$

Is this a good idea?

Density-Based Classifiers

You want to predict output Y which has arity n_Y and values $v_1, v_2, \ldots, v_{n_y}$.

- Assume there are m input attributes called $X_1, X2, \ldots, X_m$
- Break the dataset into n_Y smaller datasets called $DS_1, DS_2, \ldots, DS_{n_y}$
- Define $DS_i =$ Records in which$Y = v_i$
- For each DS_i learn the Density Estimator M_i to model the input distribution among the $Y = v_i$ records
  - M_i estimates $P(X_1, X_2, \ldots, X_m | Y = v_i)$

Idea 2:

When you get a new set of input values $(X_1 = u_1, X_2 = u_2, \ldots, X_m = u_m)$ predict the value of Y that makes $P(Y = v_i | X_1, X_2, \ldots, X_m)$ most likely

$$\hat{Y} = \arg\max_{v_i} P(Y = v_i | X_1, X_2, \ldots, X_m)$$

Terminology

According to the probability we want to maximize

- MLE (Maximum Likelihood Estimator):

$$\hat{Y} = \arg\max_{v_i} P(X_1, X_2, \ldots, X_m | Y = v_i)$$

- MAP (Maximum A-Posteriori Estimator):

$$\hat{Y} = \arg\max_{v_i} P(Y = v_i | X_1, X_2, \ldots, X_m)$$

Terminology

According to the probability we want to maximize

- MLE (Maximum Likelihood Estimator):

$$\hat{Y} = \arg\max_{v_i} P(X_1, X_2, \ldots, X_m | Y = v_i)$$

- MAP (Maximum A-Posteriori Estimator):

$$\hat{Y} = \arg\max_{v_i} P(Y = v_i | X_1, X_2, \ldots, X_m)$$

We can compute the second by applying the Bayes Theorem:

$$\begin{aligned} P(Y = v_i | X_1, X_2, \ldots, X_m) &= \frac{P(X_1, X_2, \ldots, X_m | Y = v_i) P(Y = v_i)}{P(X_1, X_2, \ldots, X_m)} \\ &= \frac{P(X_1, X_2, \ldots, X_m | Y = v_i) P(Y = v_i)}{\sum_{j=0}^{n_Y} P(X_1, X_2, \ldots, X_m | Y = v_j) P(Y = v_j)} \end{aligned}$$

Bayes Classifiers

Using the MAP estimation, we get the Bayes Classifier:

- Learn the distribution over inputs for each value Y
  - This gives $P(X_1, X_2, \ldots, X_m | Y = v_i)$
- Estimate $P(Y = v_i)$ as fraction of records with $Y = v_i$
- For a new prediction:

$$\begin{aligned} \hat{Y} &= \arg\max_{v_i} P(Y = v_i | X_1, X_2, \ldots, X_m) \\ &= \arg\max_{v_i} P(X_1, X_2, \ldots, X_m | Y = v_i) P(Y = v_i) \end{aligned}$$

Bayes Classifiers

Using the MAP estimation, we get the Bayes Classifier:

- Learn the distribution over inputs for each value Y
  - This gives $P(X_1, X_2, \ldots, X_m | Y = v_i)$
- Estimate $P(Y = v_i)$ as fraction of records with $Y = v_i$
- For a new prediction:

$$\begin{aligned}\hat{Y} &= \arg\max_{v_i} P(Y = v_i | X_1, X_2, \ldots, X_m) \\ &= \arg\max_{v_i} P(X_1, X_2, \ldots, X_m | Y = v_i) P(Y = v_i)\end{aligned}$$

You can plug any density estimator to get your flavor of Bayes Classifier:

- Joint Density Estimator
- Naïve Density Estimator
- ...

Joint Density Bayes Classifier

In the case of the Joint Density Bayes Classifier

$$\hat{Y} = \arg\max_{v_i} P(X_1, X_2, \ldots, X_m | Y = v_i) P(Y = v_i)$$

This degenerates to a very simple rule:

$\hat{Y}$ = *the most common value of* Y *among records in which*
$X_1 = u_1, X_2 = u_2, \ldots, X_m = u_m$

Joint Density Bayes Classifier

In the case of the Joint Density Bayes Classifier

$$\hat{Y} = \arg\max_{v_i} P(X_1, X_2, \ldots, X_m | Y = v_i) P(Y = v_i)$$

This degenerates to a very simple rule:

$\hat{Y}$ = *the most common value of* Y *among records in which*
$X_1 = u_1, X_2 = u_2, \ldots, X_m = u_m$

Note: if no records have the exact set of inputs
$X_1 = u_1, X_2 = u_2, \ldots, X_m = u_m$,
then $P(X_1, X_2, \ldots, X_m | Y = v_i) = 0$ for all values of Y.

In that case we just have to guess Y's value!

Naïve Bayes Classifier

In the case of the Naïve Bayes Classifier

$$\hat{Y} = \arg\max_{v_i} P(X_1, X_2, \ldots, X_m | Y = v_i) P(Y = v_i)$$

Can be simplified in:

$$\hat{Y} = \arg\max_{v_i} P(Y = v_i) \prod_{j=0}^{m} P(X_j = u_j | Y = v_i)$$

Naïve Bayes Classifier

In the case of the Naïve Bayes Classifier

$$\hat{Y} = \arg\max_{v_i} P(X_1, X_2, \ldots, X_m | Y = v_i) P(Y = v_i)$$

Can be simplified in:

$$\hat{Y} = \arg\max_{v_i} P(Y = v_i) \prod_{j=0}^{m} P(X_j = u_j | Y = v_i)$$

Technical Hint:
If we have 10,000 input attributes the product will underflow in floating point math, so we should use logs:

$$\hat{Y} = \arg\max_{v_i} \left(\log P(Y = v_i) + \sum_{j=0}^{m} \log P(X_j = u_j | Y = v_i) \right)$$

Bayes Classifiers Summary

We have seen two class of Bayes Classifiers, but we still have to talk about:

- Many other density estimators can be slotted in
- Density estimation can be performed with real-valued inputs
- Bayes Classifiers can be built with both real-valued and discrete input

We'll see that soon!

Bayes Classifiers Summary

We have seen two class of Bayes Classifiers, but we still have to talk about:

- Many other density estimators can be slotted in
- Density estimation can be performed with real-valued inputs
- Bayes Classifiers can be built with both real-valued and discrete input

<u>We'll see that soon!</u>

A couple of Notes on Bayes Classifiers

1. Bayes Classifiers don't try to be maximally discriminative, they merely try to honestly model what's going on.
2. Zero probabilities are painful for Joint and Naïve. We can use "Dirichlet Prior" to regularize them.

<u>Not sure we'll see that in this class.</u>

Probability for Dataminers

– Probability Densities –

Dealing with Real-Valued Attributes

Real-valued attributes occur, at least, in the 50% of database records:

- Can't always quantize them
- Need to describe where they come from
- Reason about reasonable values and ranges
- Find correlations in multiple attributes

Dealing with Real-Valued Attributes

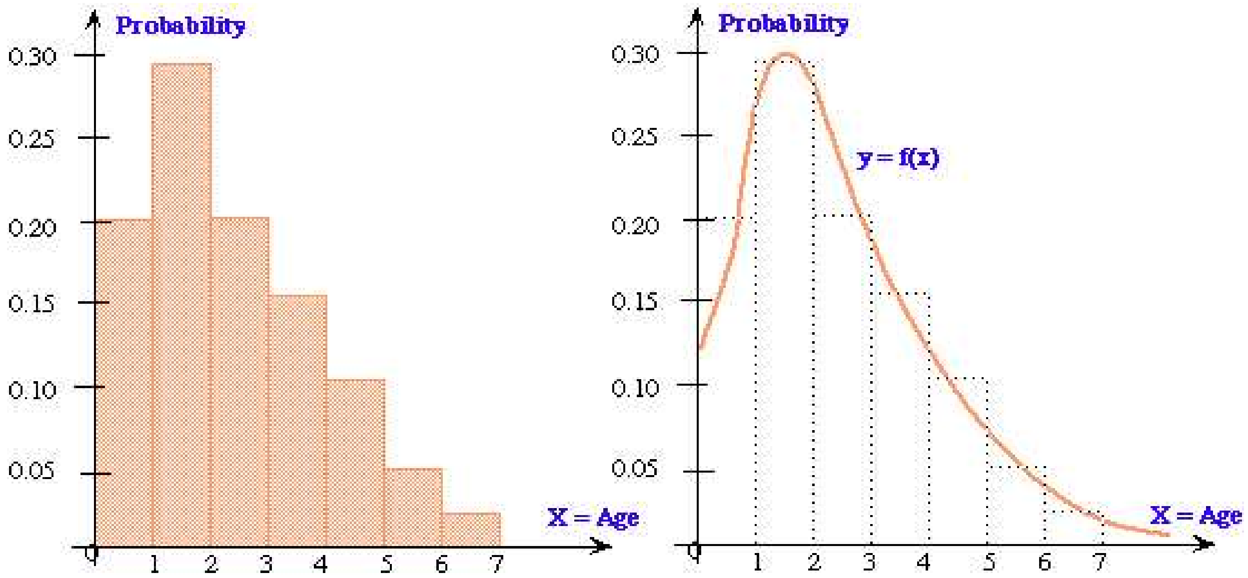
Real-valued attributes occur, at least, in the 50% of database records:

- Can't always quantize them
- Need to describe where they come from
- Reason about reasonable values and ranges
- Find correlations in multiple attributes

Why should we care about probability densities for real-valued variables?

- We can directly use Bayes Classifiers also with real-valued data
- They are the basis for linear and non-linear regression
- We'll need them for:
  - Kernel Methods
  - Clustering with Mixture Models
  - Analysis of Variance

Probability Density Function

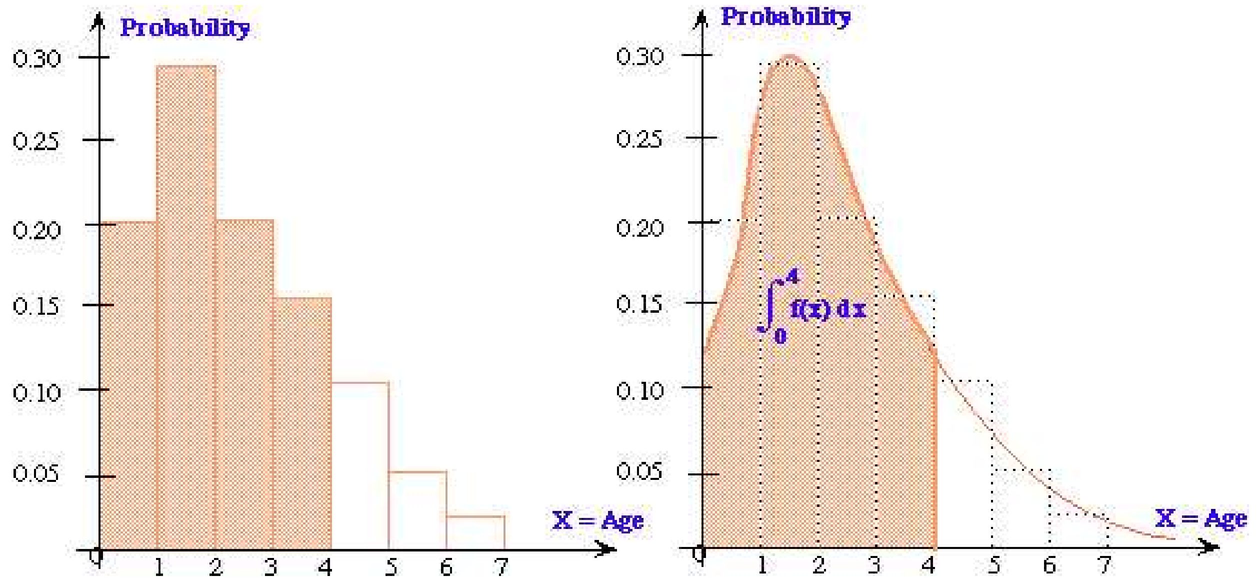


Age	Prob.
0-1	0.20
1-2	0.28
2-3	0.20
3-4	0.15
4-5	0.10
5-6	0.05
6-7	0.02

The Probability Density Function $p(x)$ for a continuous random variable X is defined as:

$$p(x) = \lim_{h \to 0} \frac{P(x - h/2 < X \leq x + h/2)}{h} \quad \longrightarrow \quad p(x) = \frac{\partial}{\partial x} P(X \leq x)$$

Properties of the Probability Density Function

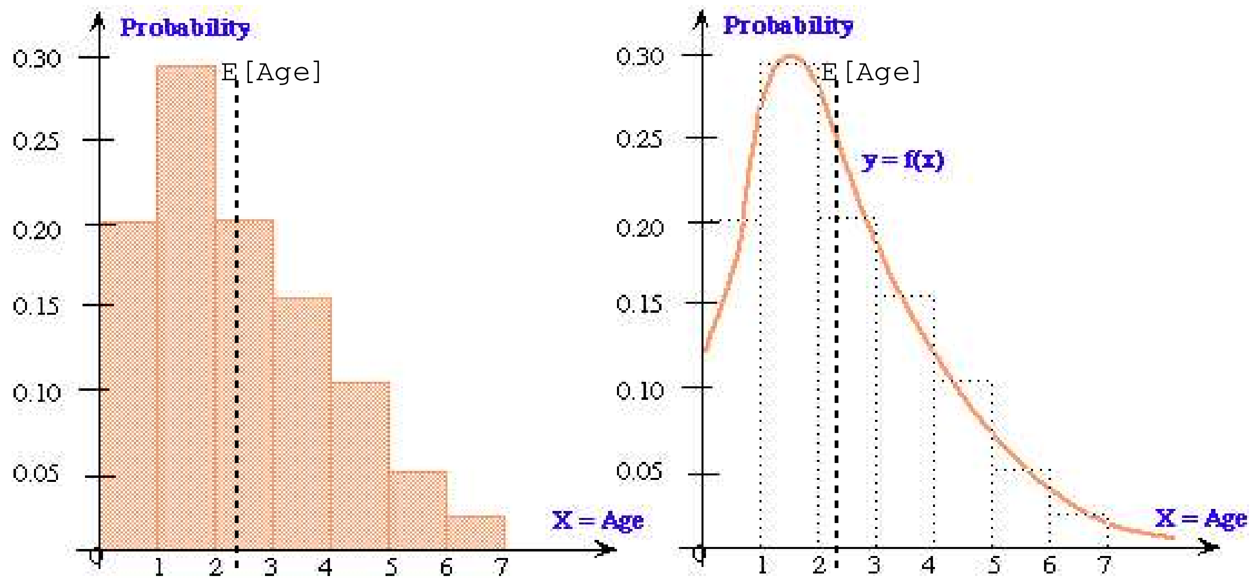


Age	Prob.
0–1	0.20
1–2	0.28
2–3	0.20
3–4	0.15
4–5	0.10
5–6	0.05
6–7	0.02

We can derive some properties of the Probability Density Function $p(x)$:

- $P(a < X \leq b) = \int_{x=a}^{b} p(x)dx$
- $\int_{x=-\infty}^{\infty} p(x)dx = 1$
- $\forall x : \quad p(x) \geq 0$

Expectation of X



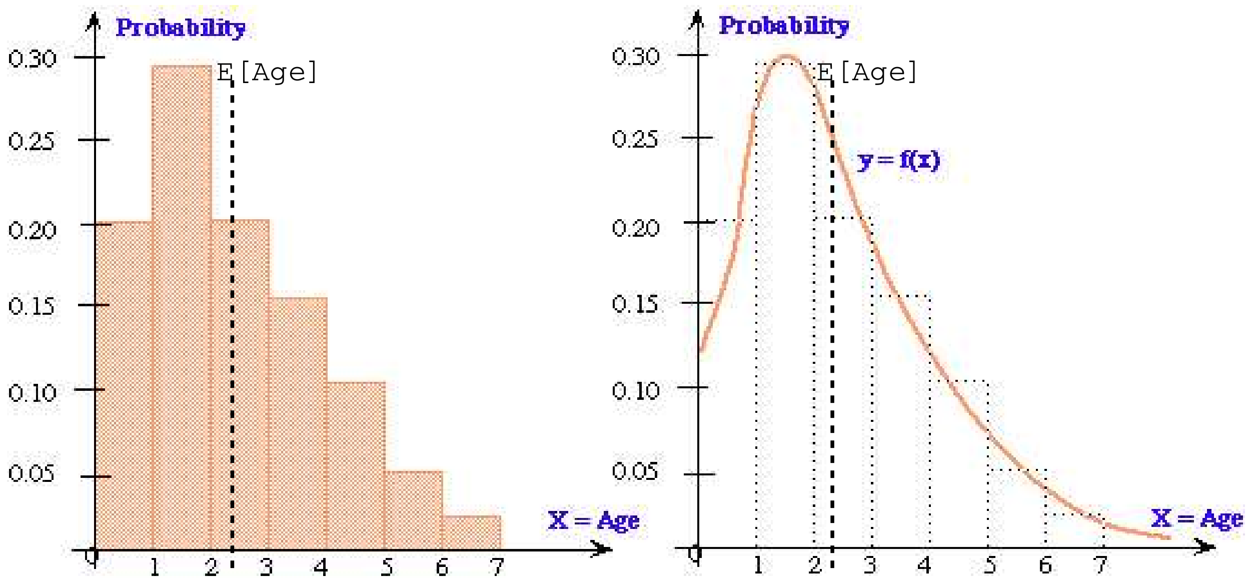
Age	Prob.
0–1	0.20
1–2	0.28
2–3	0.20
3–4	0.15
4–5	0.10
5–6	0.05
6–7	0.02

We can compute the Expectation $E[x]$ of $p(x)$:

- The average value we'd see if we look a very large number of samples of X

$$E[x] = \int_{x=-\infty}^{\infty} x\, p(x) dx = \mu$$

Variance of X



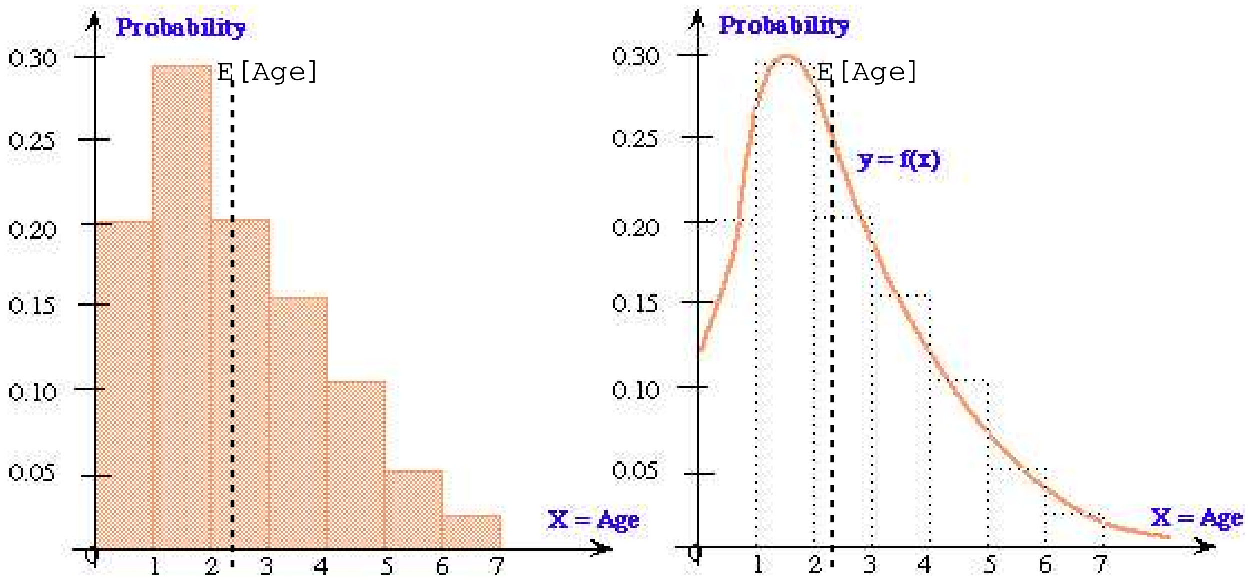
Age	Prob.
0-1	0.20
1-2	0.28
2-3	0.20
3-4	0.15
4-5	0.10
5-6	0.05
6-7	0.02

We can compute the Variance $Var[x]$ of $p(x)$:

- The expected squared difference between x and $E[x]$

$$Var[x] = \int_{x=-\infty}^{\infty} (x - \mu)^2 \, p(x) dx = \sigma^2$$

Standard Deviation of X



Age	Prob.
0–1	0.20
1–2	0.28
2–3	0.20
3–4	0.15
4–5	0.10
5–6	0.05
6–7	0.02

We can compute the Standard Deviation $STD[x]$ of $p(x)$:

- The expected difference between x and $E[x]$

$$STD[x] = \sqrt{Var[x]} = \sigma$$

Probability Density Functions in 2 Dimensions

Let X, Y be a pair of continuous random variables, and let R be some region of (X, Y) space:

$$p(x,y) = \lim_{h \to 0} \frac{P(x - h/2 < X \le x + h/2) \wedge P(y - h/2 < Y \le y + h/2)}{h^2}$$

$$P((X,Y) \in R) = \int \int_{(X,Y) \in R} p(x,y)\, dy\, dx$$

$$\int_{x=-\infty}^{\infty} \int_{y=-\infty}^{\infty} p(x,y)\, dy\, dx = 1$$

You can generalize to m dimensions

$$P((X_1, X_2, \ldots, X_m) \in R) = \int \int_{(X,Y) \in R} \ldots \int p(x_1, x_2, \ldots, x_m)\, dx_m \ldots dx_2\, dx_1$$

Marginalization, Independence, and Conditioning

It is possible to get the projection of a multivariate density distribution through Marginalization:

$$p(x) = \int_{y=-\infty}^{\infty} p(x,y)\, dy$$

If X and Y are Independent then knowing the value of X does not help predict the value of Y

$$X \perp Y \text{ iff } \forall\, x,y:\; p(x,y) = p(x)p(y)$$

Defining the Conditional Distribution $p(x|y) = \frac{p(x,y)}{p(y)}$ we can derive:

$$\begin{aligned} \forall\, x,y:\; p(x,y) &= p(x)p(y) \\ \forall\, x,y:\; p(x|y) &= p(x) \\ \forall\, x,y:\; p(y|x) &= p(y) \end{aligned}$$

Multivariate Expectation and Covariance

We can define Expectation also for multivariate distributions:

$$\mu_{\mathbf{X}} = E[\mathbf{X}] = \int \mathbf{x}\, p(\mathbf{x}) d\mathbf{x}$$

Let $X = (X_1, X_2, \ldots, X_m)$ be a vector of m continuous random variables we define Covariance:

$$\mathbf{S} = Cov[\mathbf{X}] = E[(\mathbf{X} - \mu_{\mathbf{X}})(\mathbf{X} - \mu_{\mathbf{X}})^T]$$

$$\mathbf{S}_{ij} = Cov[X_i, X_j] = \sigma_{ij}$$

- S is a $k \times k$ symmetric non-negative definite matrix
- If all distributions are linearly independent it is positive definite
- If the distributions are linearly dependent it has determinant zero

Probability for Dataminers

– Gaussian Distribution –

Gaussian Distribution Intro

We are going to review a very common piece of Statistics:

- We need them to understand Bayes Optimal Classifiers
- We need them to understand regression
- We need them to understand neural nets
- We need them to understand mixture models
- . . .

Gaussian Distribution Intro

We are going to review a <u>very</u> common piece of Statistics:

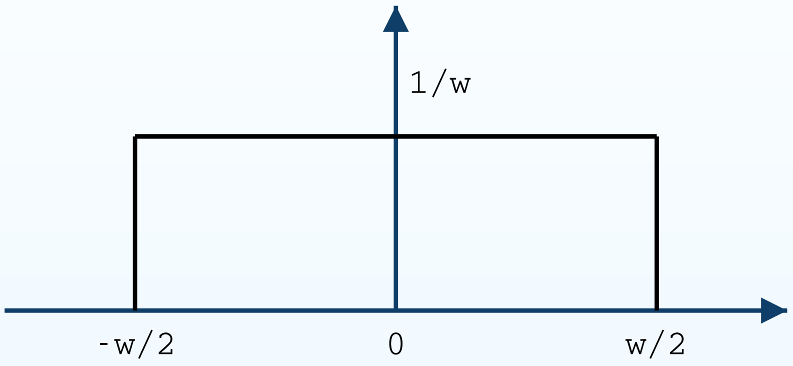
- We need them to understand Bayes Optimal Classifiers
- We need them to understand regression
- We need them to understand neural nets
- We need them to understand mixture models
- ...

Just recall before starting: the larger the entropy of a distribution ...

- ...the harder it is to predict
- ...the harder it is to compress it
- ...the less spiky the distribution

The “Box” Distribution

$$p(x) = \begin{cases} \frac{1}{w} & \text{if } |x| \leq \frac{w}{2} \\ 0 & \text{if } |x| > \frac{w}{2} \end{cases}$$

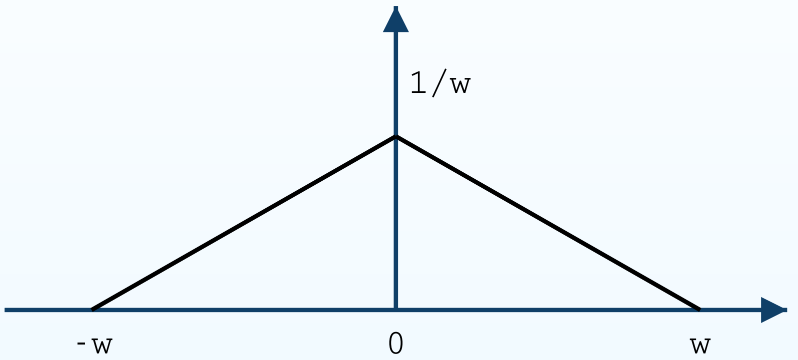


For this particular case of Uniform Distribution we have:

$$\begin{aligned} E[X] &= 0 \text{ and } Var[X] = \frac{w^2}{12} \\ H[X] &= -\int_{-\infty}^{\infty} p(x) \log p(x)\, dx = -\int_{-w/2}^{w/2} \frac{1}{w} \log \frac{1}{w}\, dx = \\ &= -\frac{1}{w} \log \frac{1}{w} \int_{-w/2}^{w/2} dx = \log w \end{aligned}$$

The "Hat" Distribution

$$p(x) = \begin{cases} \frac{w-|x|}{w^2} & \text{if } |x| \le w \\ 0 & \text{if } |x| > w \end{cases}$$



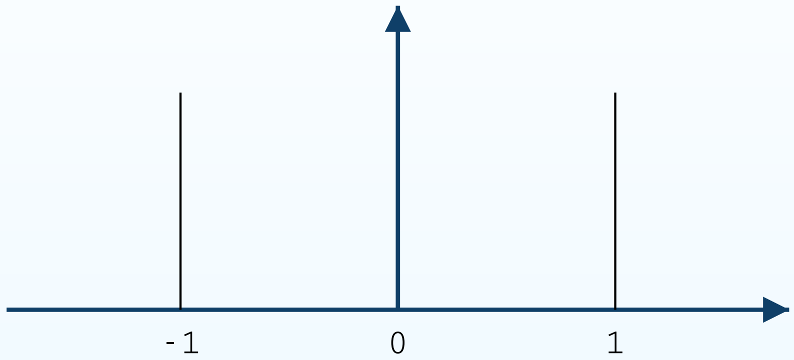
For this distribution we have:

$$E[X] = 0 \text{ and } Var[X] = \frac{w^2}{6}$$

$$H[X] = -\int_{-\infty}^{\infty} p(x) \log p(x)\, dx = \ldots$$

The “Two Spikes” Distribution

$$p(x) = \frac{\delta(x=-1) + \delta(x=1)}{2}$$

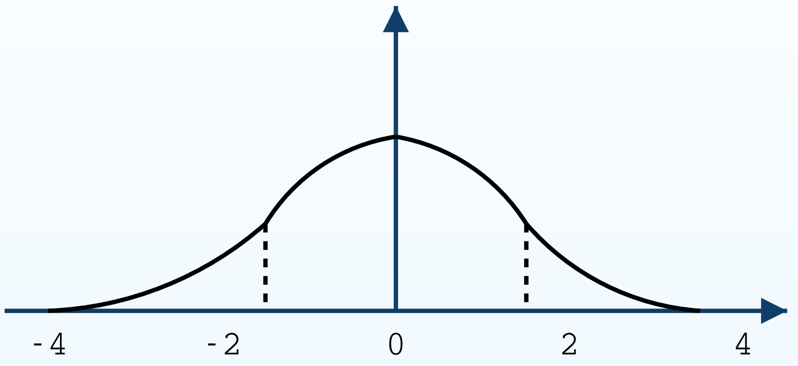


For this distribution we have:

$$E[X] = 0 \text{ and } Var[X] = 1$$
$$H[X] = -\int_{-\infty}^{\infty} p(x) \log p(x)\, dx = -\infty$$

The Gaussian Distribution

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



For this distribution we have:

$$\begin{aligned} E[X] &= \mu \text{ and } Var[X] = \sigma^2 \\ H[X] &= -\int_{-\infty}^{\infty} p(x) \log p(x)\, dx = \ldots \end{aligned}$$

"Why Should We Care About Gaussian Distribution?"

1. Largest possible entropy of any unit-variance distribution
   - "Box" Distribution: $H(X) = 1.242$
   - "Hat" Distribution: $H(X) = 1.396$
   - "Two Spikes" Distribution: $H(X) = -\infty$
   - "Gauss" Distribution: $H(X) = 1.4189$

“Why Should We Care About Gaussian Distribution?”

1. Largest possible entropy of any unit-variance distribution
   - “Box” Distribution: $H(X) = 1.242$
   - “Hat” Distribution: $H(X) = 1.396$
   - “Two Spikes” Distribution: $H(X) = -\infty$
   - “Gauss” Distribution: $H(X) = 1.4189$

2. The Central Limit Theorem
   - If $(X_1, X_2, \ldots, X_N)$ are i.i.d. continuous random variables
   - Define $z = f(x_1, x_2, \ldots, x_N) = \frac{1}{N}\sum_{n=1}^{N} x_n$
   - As $N \to \infty$ we obtain:

$$
\begin{array}{rcl}
p(z) & \sim & N(\mu_z, \sigma_z^2) \\
\mu_z = E[X_i], & & \sigma_z^2 = Var[Xi])
\end{array}
$$

Somewhat of a justification for assuming Gaussian noise!

Multivariate Gaussians

We can define gaussian distributions also in higher dimensions:

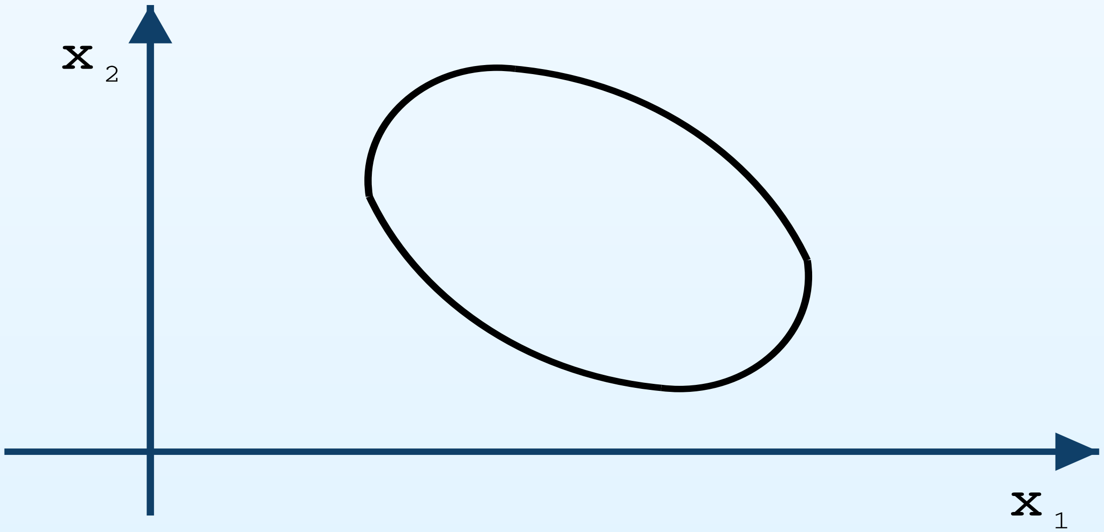
$$\mathbf{X} = \begin{pmatrix} X_1 \\ X_2 \\ \dots \\ X_m \end{pmatrix} \qquad \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \dots \\ \mu_m \end{pmatrix} \qquad \mathbf{S} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1m} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{m1} & \sigma_{m2} & \cdots & \sigma_m^2 \end{pmatrix}$$

Thus obtaining that $\mathbf{X} \sim N(\mathbf{x}, \mathbf{S})$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{m/2}||\mathbf{S}||^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \mathbf{S}^{-1} (\mathbf{x} - \mu)\right)$$

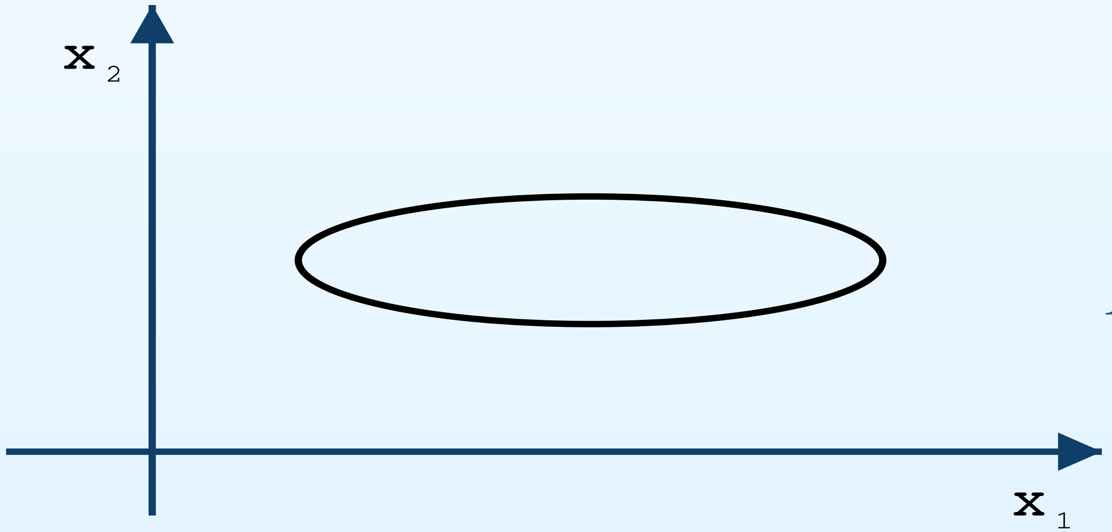
Gaussians: General Case

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \dots \\ \mu_m \end{pmatrix} \qquad \mathbf{S} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1m} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{m1} & \sigma_{m2} & \cdots & \sigma_m^2 \end{pmatrix}$$



Gaussians: Axis Alligned

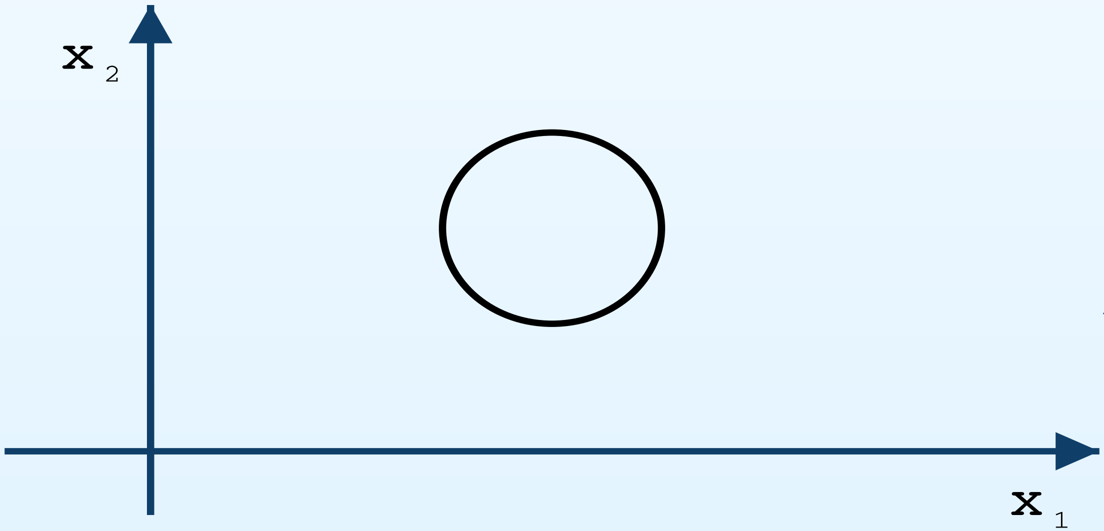
$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \dots \\ \mu_m \end{pmatrix} \qquad \mathbf{S} = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_m^2 \end{pmatrix}$$



$$X_i \perp Y_j \quad \forall \quad i \neq j$$

Gaussians: Axis Spherical

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \dots \\ \mu_m \end{pmatrix} \qquad \mathbf{S} = \begin{pmatrix} \sigma^2 & 0 & \cdots & 0 \\ 0 & \sigma^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma^2 \end{pmatrix}$$



$$X_i \perp Y_j \quad \forall \quad i \neq j$$