Methods for Intelligent Systems

Lecture Notes on Machine Learning

Matteo Mattecci

matteucci@elet.polimi.it

Department of Electronics and Information

Politecnico di Milano

Unsupervised Learning

– Density Estimation –

The world is a very unceratin place!

Thus there have been attempts to use different methodologies for dealing with world uncertainty:

- Probability theory ← **We will focus on Probabilistic Modelig!**
- Fuzzy logic
- Dempster-Shafer
- Non-monotonic reasoning

A probabilistic **model of the data** can be used to:

- Make inference about missing inputs
- Generate prediction/fantasies/imagery
- Make decisions which minimise expected loss
- Communicate the data in an efficient way

Statistical modeling is equivalent to Information Theoretic Learning (finding compact representations of the data).

The Joint Distribution

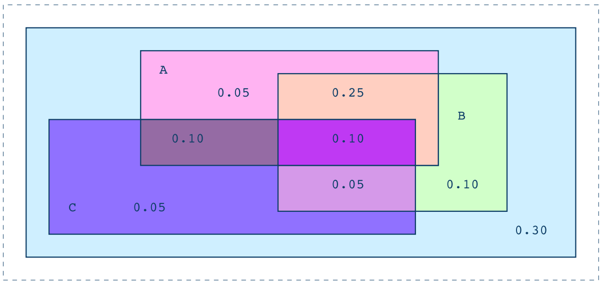
Given two random variables X and Y, the joint distribution of X and Y is the distribution of X and Y together: $P(X, Y)$.

How to make a joint distribution of M variables:

1. Make a truth table listing all combination of values
2. For each combination state/compute how probable it is
3. Check that all probabilities sum up to 1

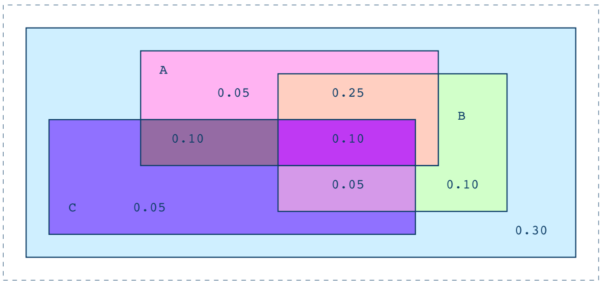
Example with 3 boolean variables A, B and C.

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10



Using the Joint Distribution (I)

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10



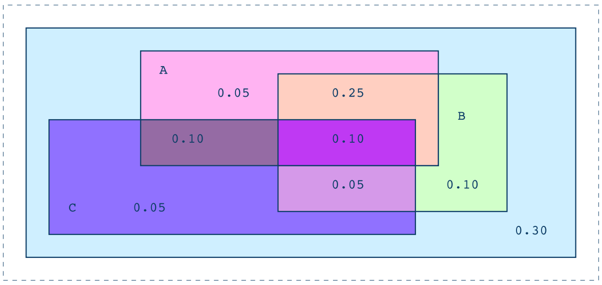
Compute **probability for logic expression**: $P(E) = \sum_{Row \sim E} P(Row)$.

- $P(A) = 0.05 + 0.10 + 0.25 + 0.10 = 0.5$
- $P(A \land B) = 0.25 + 0.10 = 0.35$
- $P(\bar{A} \lor C) = 0.30 + 0.05 + 0.10 + 0.05 + 0.05 + 0.25 = 0.8$

Can't we do something more useful?

Using the Joint Distribution (II)

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10



Use it for **making inference**: $P(E_1|E_2) = \frac{P(E_1 \land E_2)}{P(E_2)} = \frac{\sum_{Row \sim E_1 \land E_2} P(Row)}{\sum_{Row \sim E_2} P(Row)}$.

- $P(A|B) = (0.25 + 0.10)/(0.10 + 0.05 + 0.25 + 0.10) = 0.35/0.50 = 0.70$
- $P(C|A \land B) = (0.10)/(0.25 + 0.10) = 0.10/0.35 = 0.285$
- $P(\bar{A}|C) = (0.05 + 0.05)/(0.05 + 0.05 + 0.10 + 0.10) = 0.10/0.30 = 0.333$

Where do we get the Joint Density from?

Setting up a Joint Distribution

Now we know what they are and how to use them, but where do Joint Distributions come from?

- Human experts
- Simpler probabilistic facts and some algebra
  - Suppose you knew

$$\begin{aligned} P(A) &= 0.7 \\ P(B|A) &= 0.2 \\ P(B|\sim A) &= 0.1 \\ P(C|A\wedge B) &= 0.1 \end{aligned} \qquad \begin{aligned} P(C|A\wedge \sim B) = 0.8 \\ P(C|\sim A\wedge B) = 0.3 \\ P(C|\sim A\wedge \sim B) = 0.1 \end{aligned}$$

  - Then you can automatically compute the JD using the chain rule
- Learn them from data!

You should already know about density estimation, isn't it?

The Joint Distribution Estimator

A Density Estimator learns a mapping from a set of attributes to a probability distribution over the attributes space

Our Joint Distribution learner is our first example of something called Density Estimation

- Build a Joint Distribution table for your attributes in which the probabilities are unspecified
- The fill in each row with

$$\hat{P}(\text{row}) = \frac{\text{records matching row}}{\text{total number of records}}$$

We will come back to its formal definition at the end of this lecture don't worry, but now . . .

How can we evaluate it?

Evaluating a Density Estimator

We can use **likelihood** for evaluating density estimation:

- Given a record $\mathbf{x}$, a density estimator M tells you how likely it is

$$\hat{P}(\mathbf{x}|M)$$

- Given a dataset with R records, a density estimator can tell you how likely the dataset is under the assumption that all records were **independently** generated from it

$$\hat{P}(\text{dataset}) = \hat{P}(\mathbf{x}_1 \wedge \mathbf{x}_2 \wedge \ldots \wedge \mathbf{x}_R|M) = \prod_{k=1}^{R} \hat{P}(\mathbf{x}_k|M)$$

Since likelihood can get too small we usually use **log-likelihood**:

$$\log \hat{P}(\text{dataset}) = \log \prod_{k=1}^{R} \hat{P}(\mathbf{x}_k|M) = \sum_{k=1}^{R} \log \hat{P}(\mathbf{x}_k|M)$$

Joint Distribution Summary

Now we have a way to learn a Joint Density estimator from data

- Joint Density estimators can do many **good** things
  - Can sort the records by probability, and thus spot weird records (e.g., anomaly/outliers detection)
  - Can do inference: $P(E_1|E_2)$ (e.g., Automatic Doctor, Help Desk)
  - Can be used for Bayes Classifiers (see later)
- Joint Density estimators can **badly** overfit!
  - Joint Estimator just mirrors the training data
  - Suppose you see a **new dataset**, its likelihood is going to be:

$$\log \hat{P}(\text{new dataset}|M) = \sum_{k=1}^{R} \log \hat{P}(\mathbf{x}_k|M) = -\infty$$
$$if\ \exists k: \ \hat{P}(\mathbf{x}_k|M) = 0$$

We need something which generalizes! → **Naïve Density Estimator**

Naïve Density Estimator

The naïve model assumes that each attribute is distributed independently of any of the other attributes.

- Let $\mathbf{x}[i]$ denote the i^{th} field of record $\mathbf{x}$.
- The Naïve Density Estimator says that:

$$\mathbf{x}[i] \perp \{\mathbf{x}[1], \mathbf{x}[2], \ldots, \mathbf{x}[i-1], \mathbf{x}[i+1], \ldots, \mathbf{x}[M]\}$$

It is important to recall every time we use a Naïve Density that:

- Attributes are equally important
- Knowing the value of one attribute says nothing about the value of another
- Independence assumption is almost never correct . . .
- . . . this scheme works well in practice!

Naïve Density Estimator: An Example

From a Naïve Distribution you can compute the Joint Distribution:

- Suppose A, B, C, D independently distributed, $P(A \wedge \bar{B} \wedge C \wedge \bar{D}) =?$

$$\begin{aligned}
P(A \wedge \bar{B} \wedge C \wedge \bar{D}) &= P(A|\bar{B} \wedge C \wedge \bar{D})P(\bar{B} \wedge C \wedge \bar{D}) \\
&= P(A)P(\bar{B} \wedge C \wedge \bar{D}) \\
&= P(A)P(\bar{B}|C \wedge \bar{D})P(C \wedge \bar{D}) \\
&= P(A)P(\bar{B})P(C \wedge \bar{D}) \\
&= P(A)P(\bar{B})P(C|\bar{D})P(\bar{D}) = P(A)P(\bar{B})P(C)P(\bar{D})
\end{aligned}$$

Example: suppose to randomly shake a green dice and a red dice

- Dataset 1: A = red value, B = green value
- Dataset 2: A = red value, B = sum of values
- Dataset 3: A = sum of values, B = difference of values

Which of these datasets violates the naïve assumption?

Learning a Naïve Density Estimator

Suppose $\mathbf{x}[1], \mathbf{x}[2], \ldots, \mathbf{x}[M]$ are independently distributed

- Once we have the Naïve Distribution, we can construct any row of the implied Joint Distribution on demand

$$P(\mathbf{x}[1] = u_1, \mathbf{x}[2] = u_2, \ldots, \mathbf{x}[M] = u_M) = \prod_{k=1}^{M} P(\mathbf{x}[k] = u_k)$$

- We can do any inference!

But how do we learn a Naïve Density Estimator?

$$\hat{P}(\mathbf{x}[i] = u) = \frac{\text{number of record for which } \mathbf{x}[i] = u}{\text{total number of records}}$$

Please wait a few minute, I'll get the reason for this too!!

Joint Density vs. Naïve Density

What we got so far? Let's try to summarize things up:

- Joint Distribution Estimator
  - Can model anything
  - Given 100 records and more than 6 Boolean attributes will perform poorly
  - Can easily overfit the data
- Naïve Distribution Estimator
  - Can model only very boring distributions
  - Given 100 records and 10,000 multivalued attributes will be fine
  - Quite robust to overfitting

So far we have two simple density estimators, in other lectures we'll see vastly more impressive ones (Mixture Models, Bayesian Networks, . . .).

But first, why should we care about density estimation?

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 $X_1, X_2, \ldots, X_m$
- Break the dataset into n_Y smaller datasets called $DS_1, DS_2, \ldots, DS_{n_y}$
- Define DS_i as the records for which $Y = v_i$
- For each DS_i learn Density Estimator M_i to model input distribution among the $Y = v_i$ records: M_i estimates $P(X_1, X_2, \ldots, X_m | Y = v_i)$.

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:

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

- $P(Y = v_i | X_1, X_2, \ldots, X_m)$ most likely:

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

Which one do you prefer?

Maximum Likelihood vs. Maximum A Posteriory

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 Unleashed

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:

$\hat{Y} =$ most common Y among records having $X_1 = u_1, X_2 = u_2, \ldots, X_m = u_m$

Important 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:

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

Technical Hint:

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

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

The Example: “Is this a nice day to play golf?”

Outlook	Temp	Humid.	Windy	Play
sunny	85	85	false	No
sunny	80	90	true	No
overcast	83	78	false	Yes
rain	70	96	false	Yes
rain	68	80	false	Yes
rain	65	70	true	No
overcast	64	65	true	Yes
sunny	72	95	false	No
sunny	69	70	false	Yes
rain	75	80	false	Yes
sunny	75	70	true	Yes
overcast	72	90	true	Yes
overcast	81	75	false	Yes
rain	71	80	true	No

Attribute	Value	Play	Don’t
Outlook	sunny	2 (2/9)	3 (3/5)
	overcast	4 (4/9)	0 (0)
	rain	3 (3/9)	2 (2/5)
Temp.	hight	2 (2/9)	2 (2/5)
	mild	4 (4/9)	2 (2/5)
	cool	3 (3/9)	1 (2/1)
Humid.	high	3 (3/9)	4 (4/5)
	normal	6 (6/9)	1 (1/5)
Windy	true	6 (3/9)	2 (4/5)
	false	3 (6/9)	3 (1/5)

- Play = 9 (9/14)
- Don’t Play = 5 (5/14)

The Example: “A brand new day”

You wake up and gain some new *Evidence* about the day:

Outlook	Temp	Humid.	Windy	Play
sunny	cool	high	true	???

Should you play golf or not? Let’s ask it to our Naïve Bayes Classifier!

$$
\begin{aligned}
P(Evidence|yes) &= P(sunny|yes)P(cool|yes)P(high|yes)P(true|yes) \\
&= 2/9 \cdot 3/9 \cdot 3/9 \cdot 3/9 = 0.00823 \\
P(Evidence|no) &= P(sunny|no)P(cool|no)P(high|no)P(true|no) \\
&= 3/5 \cdot 1/5 \cdot 4/5 \cdot 3/5 = 0.0576 \\
P(Play = yes|Evidence) &= \frac{P(Evidence|yes)P(yes)}{\sum_{class} P(Evidence|class)P(class)} = 0.205 \\
P(Play = no|Evidence) &= \frac{P(Evidence|no)P(no)}{\sum_{class} P(Evidence|class)P(class)} = 0.795
\end{aligned}
$$

Missing Values (still “A brand new day”)

You wake up and gain some new “partial” *Evidence* about the day:

Outlook	Temp	Humid.	Windy	Play
`???`	`cool`	`high`	`true`	`???`

Apply Naïve Bayes Classifier skipping the missing values!

$$\begin{aligned}
P(Evidence|yes) &= P(cool|yes)P(high|yes)P(true|yes) \\
&= 3/9 \cdot 3/9 \cdot 3/9 = 0.037 \\
P(Evidence|no) &= P(cool|no)P(high|no)P(true|no) \\
&= 1/5 \cdot 4/5 \cdot 3/5 = 0.096 \\
P(Play = yes|Evidence) &= \frac{P(Evidence|yes)P(yes)}{\sum_{class} P(Evidence|class)P(class)} = 0.41 \\
P(Play = no|Evidence) &= \frac{P(Evidence|no)P(no)}{\sum_{class} P(Evidence|class)P(class)} = 0.59
\end{aligned}$$

The “Zero Frequency” Problem

What if an attribute value doesn’t occur with every class value (e.g. `Outlook = overcast` for class no)?

- Probability will be zero!
- No matter how likely the other values are, also a-posteriori probability will be zero!

$$P(Outlook = overcast|no) = 0 \quad \rightarrow \quad P(no|Evidence) = 0$$

The solution is related to something called “smoothing prior”:

- Add 1 to the count for every attribute value-class combination

This simple approach (Laplace estimator) solves the problem and stabilize probability estimates!

We can do also fancy things!

M-Estimate Probability

We can use **M-estimate probability** to estimate $P(Attribute|Class)$:

$$P(A|C) = \frac{n_A + mp}{n + m}$$

- n_A number of examples with class C and attribute A
- n number of examples with class C
- $p = 1/k$ whre k are the possible values of attribute A
- m is a costant value

Example: For the `Outlook` attribute we get:

`sunny`$= \frac{2+m/3}{9+m}$, `overcast`$= \frac{4+m/3}{9+m}$, `rain`$= \frac{3+m/3}{9+m}$.

We can also use weights $p_1, p_2, \ldots, p_k$ summing up to 1!

`sunny`$= \frac{2+m/p_1}{9+m}$, `overcast`$= \frac{4+m/p_2}{9+m}$, `rain`$= \frac{3+m/p_3}{9+m}$.

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!

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 Bayesian estimators with ad-hoc priors to regularize them.

Not sure we'll see that in this class.

Dealing with Real-Valued Attributes

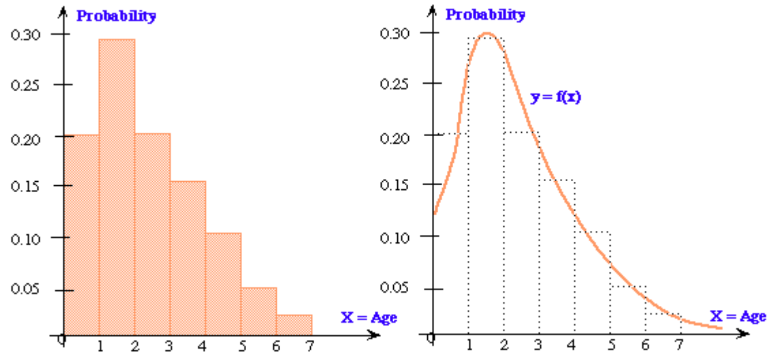
Real-valued attributes occur, at least, in the 50% of database records:

- 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

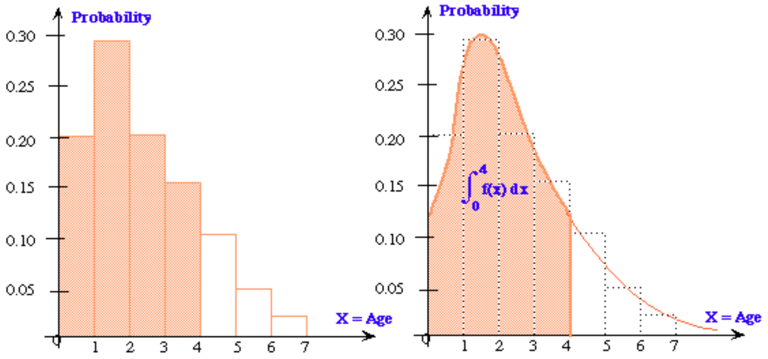


Age	Prob.
0-1	0.20
1-2	0.28
2-3	0.20
3-4	0.15
4-5	0.10
5-6	0.05
6-7	0.02

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

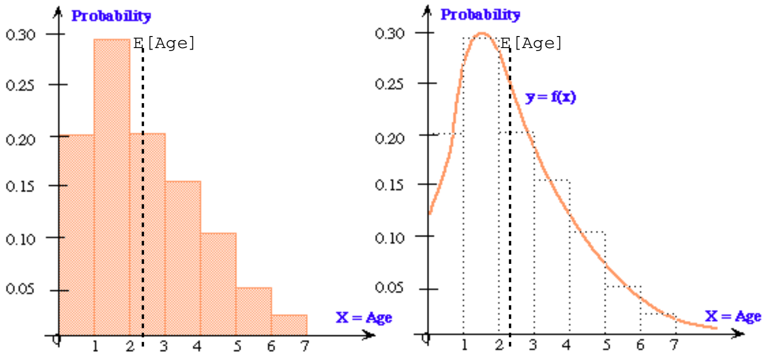


Age	Prob.
0–1	0.20
1–2	0.28
2–3	0.20
3–4	0.15
4–5	0.10
5–6	0.05
6–7	0.02

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



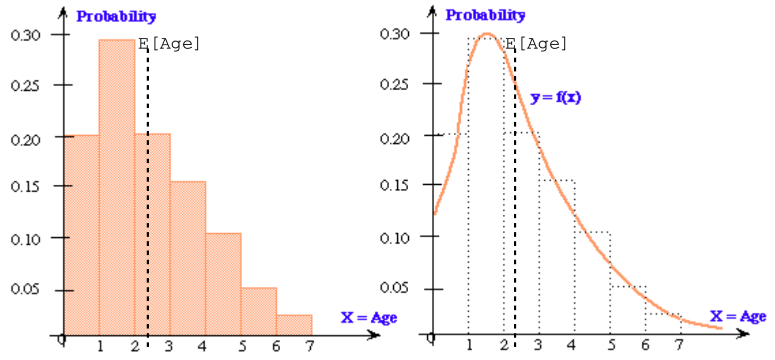
Age	Prob.
0–1	0.20
1–2	0.28
2–3	0.20
3–4	0.15
4–5	0.10
5–6	0.05
6–7	0.02

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



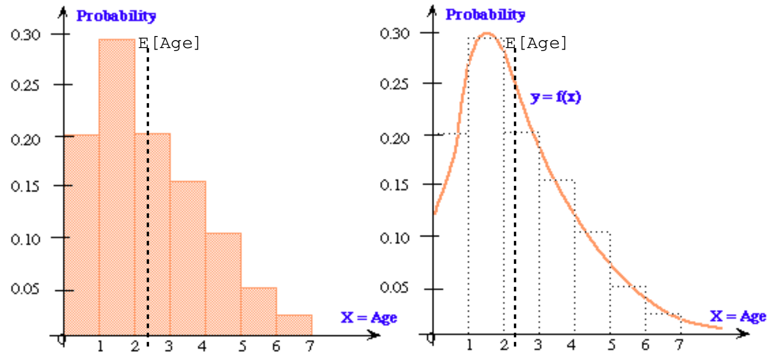
Age	Prob.
0-1	0.20
1-2	0.28
2-3	0.20
3-4	0.15
4-5	0.10
5-6	0.05
6-7	0.02

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:

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

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

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

You can generalize to m dimensions

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

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:

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

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

- 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

Gaussian Distribution Intro

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

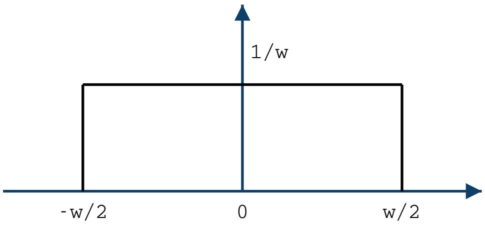
- We need them to understand Bayes Optimal Classifiers
- We need them to understand regression
- We need them to understand neural nets
- We need them to understand mixture models
- ...

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

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

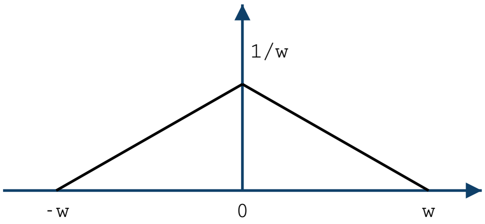


For this particular case of Uniform Distribution we have:

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

The “Hat” Distribution

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

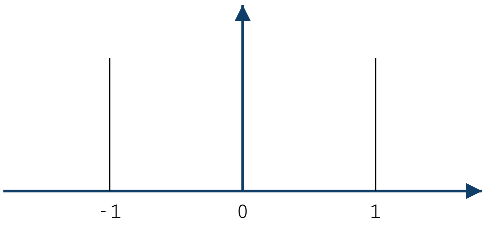


For this distribution we have:

$$\begin{aligned} E[X] &= 0 \text{ and } Var[X] = \frac{w^2}{6} \\ H[X] &= -\int_{-\infty}^{\infty} p(x) \log p(x)\, dx = \ldots \end{aligned}$$

The “Two Spikes” Distribution

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

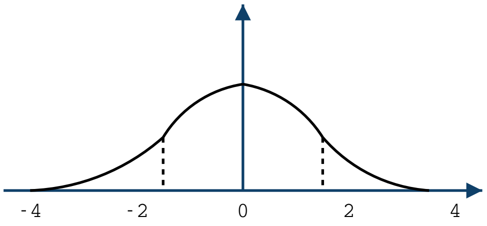


For this distribution we have:

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

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: $H(X) = 1.4189$
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:

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

Somewhat of a justification for assuming Gaussian noise!

Multivariate Gaussians

We can define gaussian distributions also in higher dimensions:

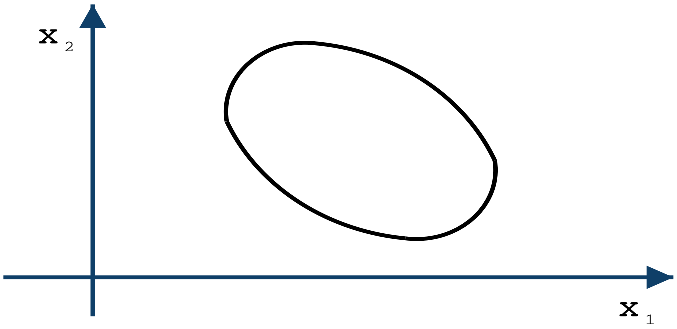
$$\mathbf{X} = \begin{pmatrix} X_1 \\ X_2 \\ \ldots \\ X_m \end{pmatrix} \qquad \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \ldots \\ \mu_m \end{pmatrix} \qquad \mathbf{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1m} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2m} \\ \vdots &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

Thus obtaining that $\mathbf{X} \sim N(\mu, \mathbf{\Sigma})$

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

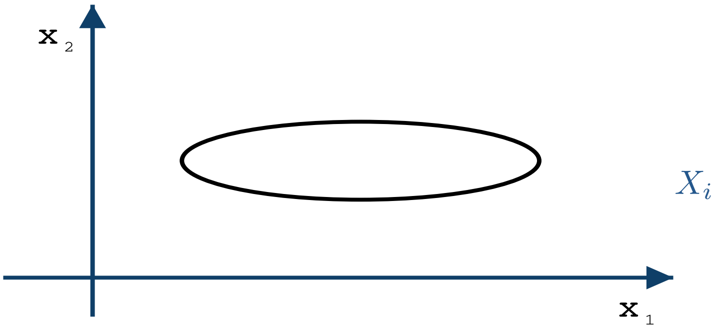
Gaussians: General Case

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



Gaussians: Axis Alligned

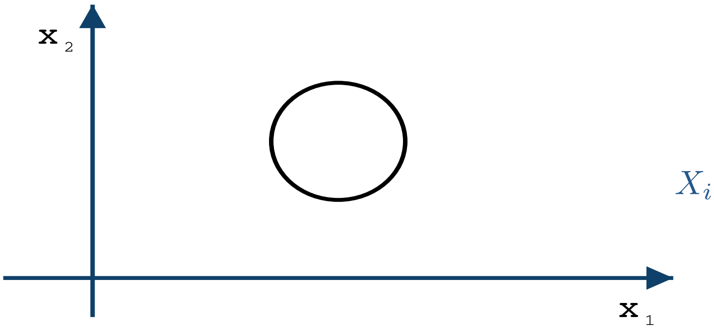
$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \dots \\ \mu_m \end{pmatrix} \qquad \mathbf{\Sigma} = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_m^2 \end{pmatrix}$$



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

Gaussians: Axis Spherical

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



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

Gaussian Bayes Classifiers

The i^{th} record in the database is created using the following algorithm:

- Generate the output (the “class”) by drawing

$$y_i \quad \sim \quad \textit{Multinomial}(p_1, p_2, \dots, p_{n_Y})$$

- Generate the inputs from a Gaussian that depends on the value of y_i:

$$\mathbf{x}_i \quad \sim \quad N_i(\mu_i, \Sigma_i).$$

Next slide outline:

1. How can you classify a new record? → Gaussian Bayes Classifier!
2. How can you build it? → Maximum Likelihood Estimation
3. How many distinct scalar parameters need to be estimated? . . .

Gaussian Bayes Classifier Explained

How can you classify a new record?

$$\begin{aligned}\hat{y} &= \arg\max_{v_i} P(y=v_i|\mathbf{x}) = \arg\max_{v_i} \frac{p(\mathbf{x}|y=v_i)p(y=v_i)}{p(\mathbf{x})} \\ &= \arg\max_{v_i} \frac{\frac{1}{(2\pi)^{m/2}||\mathbf{\Sigma_i}||^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mu_i)^T \mathbf{\Sigma}_i (\mathbf{x}-\mu_i)\right] \times p_i}{p(\mathbf{x})}\end{aligned}$$

How can you build it?

- $p_i = \frac{Num.\ Record\ with\ y=v_i}{R}$
- $\mu_i^{mle} = \frac{1}{R}\sum_{r=1\ \wedge\ y_r=v_i}^{R} x_r$
- $\mathbf{\Sigma}_i^{mle} = \frac{1}{R}\sum_{r=1\ \wedge\ y_r=v_i}^{R} (\mathbf{x}_r - \mu^{mle})(\mathbf{x}_r - \mu^{mle})^T$

How many distinct scalar parameters need to be estimated?

$$n_Y + n_Y \times (2 \times m + \frac{m(m-1)}{2}) \quad \rightarrow \quad Overfitting!$$

Gaussian Bayes Classifiers Osservations

We stiil have to face a couple of issues:

- We already had problems with Joint Bayes Classifiers for too many parameters and overfitting:
  - General gaussians: $O(n_Y) + O(n_Y \times m^2) \quad \rightarrow \quad O(n_Y \times m^2)$
  - Axis-alligned gaussians: $O(n_Y) + O(n_y \times 2m) \quad \rightarrow \quad O(n_Y \times 2m)$
  - Spherical gaussians: $O(n_Y) + O(n_Y \times m) \quad \rightarrow \quad O(n_Y \times m)$
- Mixed categorical/real density estimation ($\mathbf{u} = real, \mathbf{v} = categorical$):
  - $\mu_{i,\mathbf{v}} =$ Mean of $\mathbf{u}$ among records matching $\mathbf{v}$ and in which $y = v_i$
  - $\mathbf{\Sigma}_{i,\mathbf{v}} =$ Cov. of $\mathbf{u}$ among records matching $\mathbf{v}$ and in which $y = v_i$
  - $q_{i,\mathbf{v}} =$ Fraction of $y = v_i$ records that match $\mathbf{v}$
  - $p_i =$ Fraction of records that match $y = v_i$

$$P(Y=v_i|\mathbf{u},\mathbf{v}) = \frac{p(\mathbf{u}|\mathbf{v}, y=v_i)p(\mathbf{v}|y=v_i)P(y=v_i)}{p(\mathbf{u},\mathbf{v})} = \frac{N(\mu_{i,\mathbf{v}}, \mathbf{\Sigma}_{i,\mathbf{v}})q_{i,\mathbf{v}}p_i}{p(\mathbf{u},\mathbf{v})}$$

Naïve Gaussian Bayes Classifiers

We can apply (again) the naïve approach to reduce overfitting:

$$\begin{aligned}
P(Y = v_i|\mathbf{u}, \mathbf{v}) &= \frac{p(\mathbf{u}, \mathbf{v}|y = v_i)P(y = v_i)}{p(\mathbf{u}, \mathbf{v})} \\
&= \frac{1}{p(\mathbf{u}, \mathbf{v})} \prod_{j=1}^{q} p(\mathbf{u}[j]|\mu_i[j], \sigma^{\mathbf{2}}{}_i[j]) \times \prod_{j=q+1}^{m} P(\mathbf{v}_i[j]) \times P(y = v_i) \\
&= \frac{1}{p(\mathbf{u}, \mathbf{v})} \prod_{j=1}^{q} N(\mu_i[j], \sigma^{\mathbf{2}}{}_i[j]) \times \prod_{j=1}^{m-q} \mathbf{q}_i[j] \times p_i
\end{aligned}$$

- $\mu_i[j] =$ Mean of $\mathbf{u}[j]$ among records in which $y = v_i$
- $\sigma^{\mathbf{2}}{}_i[j] =$ Variance of $\mathbf{u}[j]$ among records in which $y = v_i$
- $\mathbf{q}_i[j] =$ Probability of $\mathbf{v}[j]$ value among records in which $y = v_i$
- $p_i =$ Fraction of records that match $y = v_i$