UNIVERSITÀ DEGLI STUDI DI TRENTO

Department of Information Engineering and Computer Science

Master Degree in
Computer Science

Machine Learning

Professor
Andrea Passerini

Assistant
Luca Erculani

Anno accademico 2019/2020

Contents

1 Introduction

A first definition of machine learning by T. Mitchell is the following one:

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

That is, machine learning builds a model that allows the computer to learn from experience, but does not solve the problem directly (as it's not always possible).
For a well-posed learning problem there a few key components:

- **Task**: For example recognising characters;
- **Performance measure**: to evaluate the learned system, that is the way the success of the predictors is measured (for example the number of misclassified characters).
- **Training experience**: to train the learning system, that is what is given to the system as experience E so that the algorithm can improve. The easiest way to do so is to give a training set made of couples.

1.1 Designing a machine learning system

The following steps should be followed when designing a machine learning system:

- Formalize the learning task;
- Collect data;
- Extract features;
- Choose class of learning models;
- Train model;
- Evaluate model.

1.1.1 Formalize the learning task

As a first step, it's important to define the task that the learning system should address, for example recognizing handwritten characters.
Often the main task can be divided in smaller tasks such that the problem gets easier to be solved. Considering the example before, it could be split in:

- Segment the image into words and then each word into characters.
- Identify the language of the writing.

- Classify each character into the language alphabet.

At last to define the task, one should also choose an appropriate performance number for evaluating such system, for example the number of misclassified characters could be a good performance measure.

1.1.2 Collect data

Since the machine learning system needs to learn from something, we need a so called training set, that is a set of example written in a *machine readable format*.
For such reason, often the data require some manual intervention, for example in labelling, in order to have a so called supervised learning, that is the training set is formed of a couple containing the initial data and also the correct result that the system should get. This, though, could be quite expansive to be achieved so a much cheaper learning mode is being used, the so called semi-supervised learning, where supervised data is mixed to data which does not have the expected result nor it's labeled.

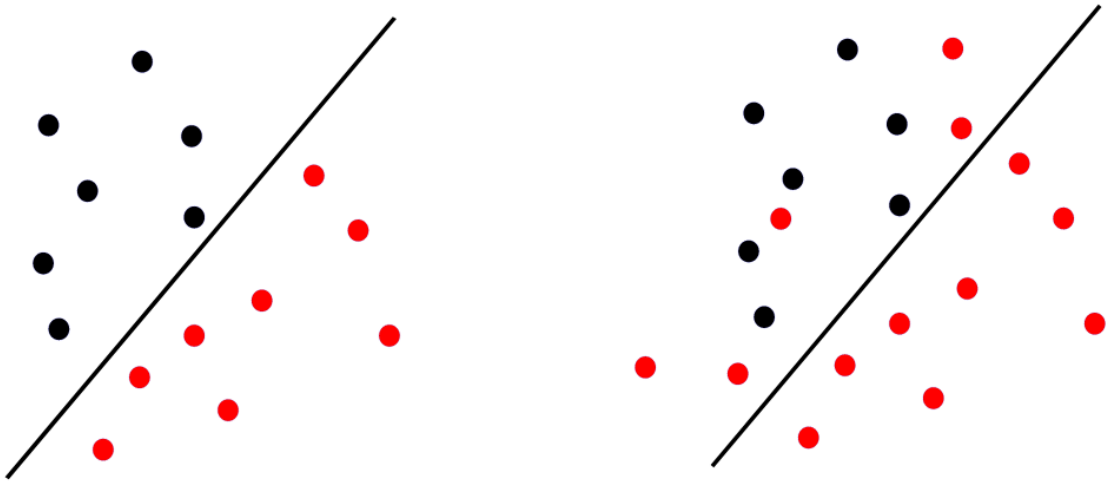
1.1.3 Extract features

Not all the data that is given is really useful and instead sometime is harmful. For this reason a relevant set of feature should be extracted from all the data, that is fundamental information for the input in order for the algorithm to give the correct solution. To do this the best way possible, some prior knowledge is usually necessary.
Mind that:

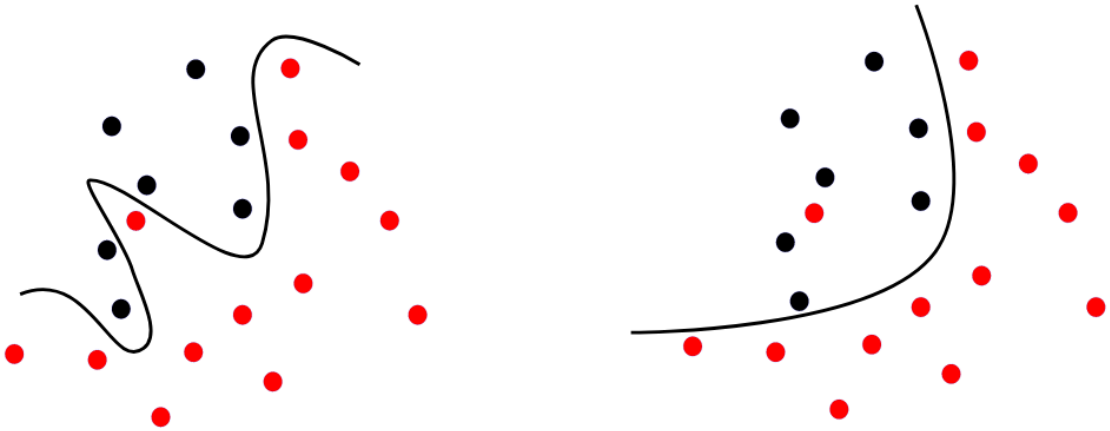
- Too few features could miss some relevant information preventing the system from learning in a correct way.
- Too many feature could heavy the system too much and the training phase could take a lot of time.
- Including noisy features complicates the training phase as the algorithm needs to learn to avoid these.

1.1.4 Model class

Many model classes are available and choosing the best one is important. For example a binary classification, that is a simple linear model that separates data in to two classes.



We call features all the points, that is the example given, and classes the colors, that is the supposed result of the algorithm. It's easy to notice that a binary classifier works as a separator that is a feature can have a class or another. Not always is possible to define a linear function that separates the features, so in some cases a more complex model can be used.



It's possible to notice that in this case the red feature in the middle of all the black ones introduce a lot fo noise. Mind that we still are in a training phase and we do not mind for some false positive o negatives, we want our system to be good in real life not in the training set. If now a complex model is designed, such as the one on the left, the system doesn't learn how to recognize noise and to get rid of it. This problem is called overfitting and should be avoided.
Quindi non vogliamo che il nostro sistema semplicemente memorizzi i dati del training, ma che riesca a generalizzarli.

1.1.5 Learning settings

We have already talked about possible types of learning, such as supervised or semi-supervised learning, but they are not all that there is.

- **Supervised learning**: the learner is given a series of pair where one element is input and one is the correct output. What the algorithm should do is to learn a function f that maps each input to output. Formally given a set of inputs $\mathcal{X}$ and outputs $\mathcal{Y}$, the learner is provided with couples $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ in order to create a *model* $f : \mathcal{X} \mapsto \mathcal{Y}$. Since each example need to have a correct output, usually a domain expert is involved in *labelling* the couples.

- **Unsupervised learning**: the learner is provided with a set of input examples $\mathcal{X}$ but without any labelling information, that is no output. In this case the learner models training examples, for instance grouping examples into clusters according to their similarity.

- **Semi-supervised learning**: this stays in the middle of the formers. As in supervised learning, the learner is provided with a set of input/output pairs: $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$. The difference is that the set of inputs is usually much larger than the outputs one providing additional unlabelled examples. As in supervised learning the goal is to find a model f that maps input into outputs: $f : \mathcal{X} \mapsto \mathcal{Y}$. The unlabelled data is exploited in order to improve performance, for example by forcing the model to produce similar outputs for similar inputs.

- **Reinforcement learning**: in this type of learning, the actor is not provided anymore with inputs and outputs, but is given a set of possible states $\mathcal{S}$ and a set of possible actions $\mathcal{A}$ that allows it to move to the next state. When the learner performs an action a from a state s, it is given a reward $r(s, a)$. The task is to learn a policy allowing to choose for each state s the action a maximizing the overall reward. One problem is that the reward may not always be immediate but instead might be delayed. If this is the case, the learner needs to find the right trade-off between exploitation and exploration. An example of delayed reward is chess where the learner knows its reward only at the end.

1.1.5.1 Choice of learning algorithm

The choice of the algorithm to use is based on the knowledge we have of the data:

- Full knowledge of probability distribution of data: Bayesian decision theory.
- Form of probabilities known, parameters unknown: parameter estimation from training data.
- Form of probabilities unknown, training examples available: do not model input data (generative methods), learn a function predicting the desired output given the input.
- Form of probabilities unknown, training examples unavailable (only inputs): unsupervised methods, cluster examples by similarity.

1.1.5.2 Learning tasks

For what concerns the supervised learning, possible tasks are:

- *Binary*: it's the easiest task, or the element belongs to a class, or to the other.
- *Multiclass*: an element can belong to one of the set of the class.
- *Multilabel*: given n possible classes, the example can be assigned to a subset of $m \leq n$ classes.
- *Regression*: in this case a real value is predicted, for example the biodegradation rate of a molecular compound.
- *Ordinal regression* (ranking): a set of examples is ordered according to their relative importance with respect to the task, for example ordering email according to their urgency.

For what concerns unsupervised learning:

- *Clustering*: data is divided into groups in order to have homogeneous elements in each group and so that each group is enough different from the others.
- *Dimensionality reduction*: in this case there are many features and the aim is to reduce their dimensionality maintaining as mach information as possible.
- *Novelty detection*: given a certain distribution of data, the aim of this task is to find all the element that are novel, that is don't respect the distribution. This approach is used for example in network traffic analysis in order to detect anomalous traffic that indicates a possible attack.

1.1.6 Train Model

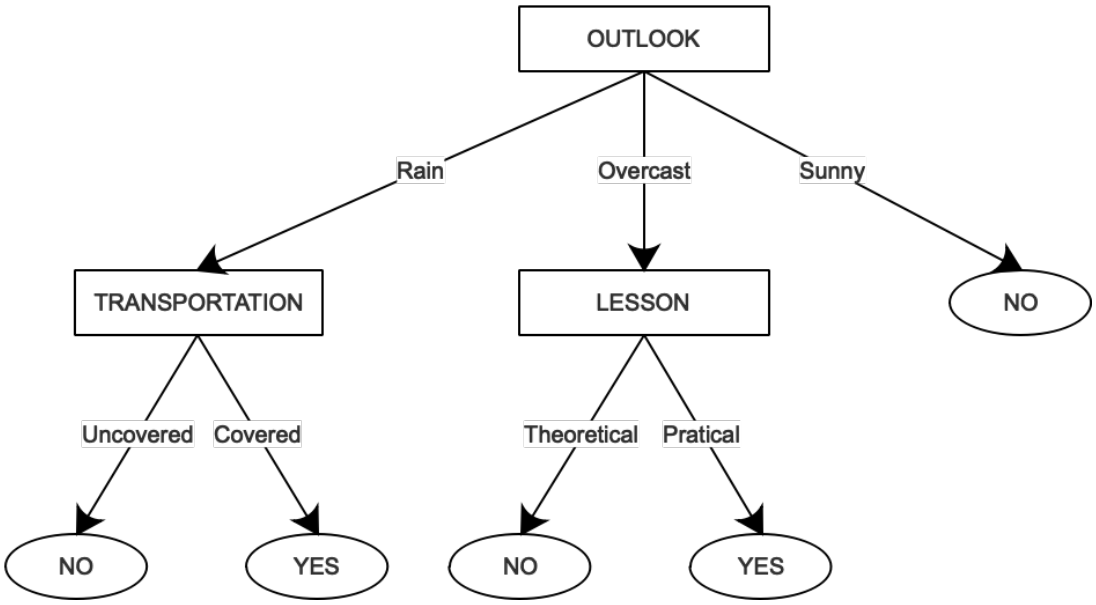
Training a model implies searching through the space of possible models (aka hypotheses) given the chosen model class.
Such search typically aims at fitting the available training examples well according to the chosen performance measure.
However, the learned model should performe well on unseen data, **generalization**, and not simply memorize training examples, **overfitting**.
Different techniques can be used to improve generalization, usually by trading off model complexity with training set fitting.

1.1.7 Evaluate Model

Once a fitting model is found, it needs to be evaluated on its ability to generalize to unseen examples.
For this reason there is the need to have a set of data that is different from the training one. Usually from a unique dataset, two are derived, one for training and one for evaluation.

2 Decision tree learning

From data is sometime possible to build a tree of features that allows to find an item's target value: given an example, with respect to the values the example assumes for each feature, it's possible to build a path which will lead to a leaf, that is a label.



A tree is a disjunction of conjunctions of constraints over attribute values, that is it express a disjunction normal form. Each path from the root to a leaf is a conjunction of hte constraints specified in the nodes along it, e.g., if it's cloudy and the lesson is theoretical, then a student is not going to go to the lesson.

$$\text{OUTLOOK} = \text{Overcast} \wedge \text{LESSON} = \text{Theoretical}$$

One of the strongest advantages of this model is its explainability: it's easy to know what the model does and what the outcome may be.
Such representation works best with binary and multiclass classification takes, even though some extensions to regression also exists, for example exploiting means properties. Another field in which it works fine is with missing attribute, that is for example when some feature don't have any value.

2.1 Algorithm

The learning algorithm in this case is simply a greedy approach proposes by Quinlan: for each node, starting from the root:

1. Choose the best attribute to be evaluated;
2. Add a child for each attribute value;
3. Split node training set into children according to value of chosen attribute;
4. Stop splitting a node if it contains examples from a single class, or there are no more attribute to test.

The idea is to start with a whole training set and then choose the feature so that the set will become as parted as possible. We stop when the example in the training set are all of the same class or there are no more features to be tested.

2.1.1 Choosing the best attribute

The best strategy to choose the best attribute on which to split, is to use sets' **homogeneity**, which is based on the concept of **entropy**.

Definition 2.1: Entropy

Entropy is a measure of the amount of information contained in a collection of instances S which can take a number c of possible values:

$$H(S) = -\sum_{i=1}^{c} p_i \log_2(p_i)$$

Where p_i is the fraction of S taking value i.

Entropy is the way in which the increment of homogeneity is measured since it's based on the information contained in a set: if all the values inside the set are the same, then there is no information, otherwise if two values are one the opposite of the other, then there is maximum entropy since the probability of taking one or the other becomes 5%.
Given the entropy then it's possible to compute the **information** fain:

Definition 2.2: Information Gain

The information gain IG is the expected reduction in entropy obtained by partitioning a set S according to the value of a certain attribute A:

$$IG(S, A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H(S_v)$$

The greater the difference (IG), the better it is.

2.1.2 Overfitting

With decision tree it's possible to end up with overfitting for example when from each set of n elements, only one element is selected at the time. In this way the leaves would just contain one example creating a complex tree and leading to overfitting.
So the goal is not to obtain pure leaves, but instead to have also leaves that can contains example not strictly from that class.
There are two ways to avoid this possibility in decision trees:

- **Pre-pruning**: decide whether to stop splitting a node even if it contains training example with different labels.
- **Post-pruning**: learn a full tree, risking overfitting, and then in case prune to remove subtrees. Usually this strategy is the one used.

Let's focus on the post-pruning strategy: for this we shall introduce a validation set, that is a set against which to test the trained model. Let's consider a full tree, then:

1. For each node in the tree: evaluate the performance on the validation set when removing the subtree rooted at it;
2. If all node removals worsen performance, then stop;
3. Choose the node whose removal has the best performance improvement;
4. Replace the subtree rooted at it with a leaf;
5. Assign to the leaf the majority label of all examples in the subtree;
6. Return to 1.

When no more improvements are obtained through pruning, then the pruning is finished.
Obviously validation, training and test set must all have the same statistics.

2.2 Problem with decision tree

2.2.1 Discretization

A first problem with decision trees are continuous-valued attributes. One easy solution is to discretize the variable in order to be used in the internal nodes tests.
Discretization implies the usage of thresholds, which can be set in order to maximize the attribute quality criterion, e.g. information gain.
The procedure to achieve this is:

- Examples are sorted according to their continuous attribute values;
- For each pair of successive examples having different labels, a *candidate threshold* is places ad the average of the two attribute values.
- For each candidate threshold, the information gain is achieved splitting examples according to how it is computed;
- The threshold producing the higher information gain is used to discretize the attribute.

2.2.1.1 Example

Let's suppose that we want to accept a work based on the age via the following table:

age	class
23	y
27	y
34	n
38	n
35	y
32	n

As a first thing, ordering data can be a great deal:

age	class
23	y
27	y
32	n
34	n
35	y
38	n

Then a threshold can be chosen, for example in the point where the class changes, even if some other examples are present afterwards, so for example age=32 can be a threshold.

2.2.2 Alternative attributes test measures

Information gain has a problem: if for example the set contained people with an id, it would place each person into a leaf, because this would make the information gain as big as possible. Information gain works fine, but in some cases it divides too much.

In order to avoid the tree from becoming too much spread, it's possible to measure the entropy with respect to the attribute value instead of the class value:

$$H_A(S) = -\sum_{v \in Values(A)} \frac{|S_v|}{|S|} \log_2 \frac{|S_v|}{|S|}$$

Bigger the entropy, bigger is the division the attribute is making of the tree.
Now instead of using the information gain, we could downweight it with respect to the new entropy and compute the **gain ratio**:

$$IGR(S, A) = \frac{IG(S, A)}{H_A(S)}$$

2.2.3 Missing values

At the beginning we said that decision trees could deal also with missing values. Let's suppose that at a certain node n we'd like to split on attribute A, but the example x of class $c(x)$ is missing its value for attribute A. Two solutions are possible:

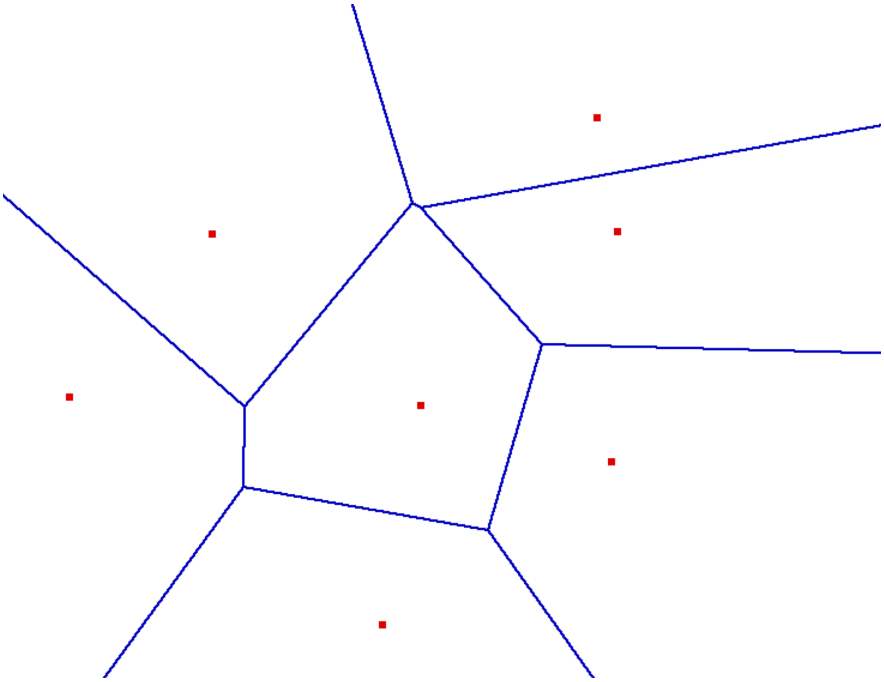
- Simple: assign to x the most common attribute values among examples in n, or the most common of examples in n with class $c(x)$. This works in every condition, but it's not very performing.
- Complex: propagate x to each of the children of n with a fractional value equal to the proportion of example with the corresponding attribute value. This implies that at test time, for each candidate class, all fractions of the test example which reached a leaf with that class are summed, and the example is assigned the class with the highest overall value.

2.3 Random Forests

The decision trees are easy to be used, but not really great in terms of accuracy. An extension of the decision trees is random forests.

Given a set of N examples, sample N examples with replacement, that is the same example can be selected multiple times. Then train a decision tree on the sample, selecting at each node m features at random among which to choose the best one. Repeat this procedure multiple times to generate a forest with multiple trees.
As for the testing, each example is tested against each tree in the forest, and the majority class among the predictions is returned.

3 k-Nearest Neighbour Learning



Definition 3.1: Metrics

Given a set $\mathcal{X}$, a function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_0^+$ is a *metric* for $\mathcal{X}$ if for any $x, y, z \in \mathcal{X}$ the following properties are satisfied:

1. **Reflixivity**: $d(x, y) = 0 \iff x = y$;
2. **Symmetry**: $d(x, y) = d(y, x)$;
3. **Triangle inequality**: $d(x, y) + d(y, z) \geq d(x, z)$

A frequently used, and well-known, function d is the Euclidean distance defined in $\mathbb{R}^n$:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

3.1 The algorithm

3.1.1 Classification problem

```
for all test examples x do
    for all training examples (x_i, y_i) do
        compute distance d(x, x_i)
    end for
    select the k-nearest neighbours of x
    return class of x as majority class among neighbours:
        argmax_y Σ_{i=1}^{k} δ(y, y_i)
end for
```

Where

$$\delta(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

3.1.2 Regression problem

```
for all test examples x do
    for all training examples (x_i, y_i) do
        compute distance d(x, x_i)
    end for
    select the k-nearest neighbours of x
    return the average output value among neighbours:
        (1/k) Σ_{i=1}^{k} y_i
end for
```

3.2 Characteristics

- **Instance-based learning**: the model used for prediction is calibrated for the test example to be processed;
- **Lazy learning**: computation is mostly deferred to the classification phase;
- **Local learner**: assumes prediction should be mainly influenced by nearby instances;
- **Uniform feature weighting**: all features are uniformly weighted in computing distances.

4 Linear Algebra

Definition 4.1: Vector Space

A set $\mathcal{X}$ is called a vector space over $\mathbb{R}$ if addition and scalar multiplication are defined and, for all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{X}$ and $\lambda, \mu \in \mathbb{R}$, satisfy the following properties:

- Addition:
  - *Associative*: $\boldsymbol{x} + (\boldsymbol{y} + \boldsymbol{z}) = (\boldsymbol{x} + \boldsymbol{y}) + \boldsymbol{z}$
  - *Commutative*: $\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{y} + \boldsymbol{x}$
  - *Identity element*: $\exists 0 \in \mathcal{X} : \boldsymbol{x} + 0 = \boldsymbol{x}$
  - *Inverse element*: $\forall \boldsymbol{x} \in \mathcal{X} \exists \boldsymbol{x'} \in \mathcal{X} : \boldsymbol{x} + \boldsymbol{x'} = 0$
- Scalar multiplication:
  - *Distributive over elements*: $\lambda(\boldsymbol{x} + \boldsymbol{y} = \lambda\boldsymbol{x} + \lambda\boldsymbol{y}$
  - *Distributive over scalars*: $(\lambda + \mu)\vec{x} = \lambda\boldsymbol{x} + \mu\boldsymbol{x}$
  - *Associative over scalars*: $\lambda(\mu\boldsymbol{x}) = (\lambda\mu)\boldsymbol{x}$
  - *Identity element*: $\exists 1 \in \mathbb{R} : 1\boldsymbol{x} = \boldsymbol{x}$

Definition 4.2: Subspace

A subspace is any non-empty subset of $\mathcal{X}$ being itself a vector space.

Definition 4.3: Linear Combination

Given n scalars $\lambda_i \in \mathbb{R}$ and n vectors $\boldsymbol{x}_i \in \mathcal{X}$, their linear combination is:

$$\sum_{i=1}^{n} \lambda_i \boldsymbol{x}_i$$

Definition 4.4: Span

The span of vectors $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ is defined as the set of their linear combinations:

$$\left\{ \sum_{i=1}^{n} \lambda_i \boldsymbol{x}_i, \lambda_i \in \mathbb{R} \right\}$$

Definition 4.5: Linear independency

A set of vectors $\boldsymbol{x}_i$ is linearly independent if none of them can be written as a linear combination of the others.

Definition 4.6: Basis

A set of vectors $\boldsymbol{x}_i$ is a basis for $\mathcal{X}$ if any element in $\mathcal{X}$ can be uniquely written as a linear combination of vectors $\boldsymbol{x}_i$.

A necessary condition for this to be true is that vectors $\boldsymbol{x}_i$ are linearly independent. All bases of $\mathcal{X}$ have the same number of elements, called the *dimension* of the vector space.

Definition 4.7: Linear Map

Given two vector spaces $\mathcal{X}, \mathcal{Z}$, a function $f : \mathcal{X} \to \mathcal{Z}$ is a linear map if all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}, \lambda \in \mathbb{R}$:

- $f(\boldsymbol{x} + \boldsymbol{y}) = f(\boldsymbol{x}) + f(\boldsymbol{y})$
- $f(\lambda \boldsymbol{x}) = \lambda f(\boldsymbol{x})$

A linear map between two finite-dimensional spaces $\mathcal{X}, \mathcal{Z}$, $\mathcal{X} \underset{f}{\to} \mathcal{Z}$ of dimensions n, m can always be written as a matrix of basis transformation:

$$M \in \mathbb{R}^{m \times n} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}$$

Let $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$ and $\{\boldsymbol{z}_1, ..., \boldsymbol{z}_m\}$ be some bases for $\mathcal{X}$ and $\mathcal{Z}$ respectively. For any $\boldsymbol{x} \in \mathcal{X}$ we have that it can be written as the linear combination of vectors in the basis, hence:

$$f(\boldsymbol{x}) = f\left(\sum_{i=1}^{n} \lambda_i \boldsymbol{x}_i \right) = \sum_{i=1}^{n} \lambda_i f(\boldsymbol{x}_i)$$

Since there is a function f that maps values from $\mathcal{X}$ to $\mathcal{Z}$, then:

$$f(\boldsymbol{x}_i) = \sum_{j=1}^{m} a_{ji} \boldsymbol{z}_i$$

Which leads to:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_i a_{ji} \boldsymbol{z}_j = \sum_{j=1}^{m} \left(\sum_{i=1}^{n} \lambda_i a_{ji} \right) \boldsymbol{z}_j = \sum_{j=1}^{m} \mu_j \boldsymbol{z}_j$$

In short:

$$M\lambda = \mu$$

Definition 4.8: Matrix Properties Transpose

Given a matrix M, the transposed matrix M^T is the matrix obtained by exchanging rows and columns. Tra transposed of the product of two matrixes M, N, is the product of the two transposed:

$$(MN)^T = N^T M^T$$

Definition 4.9: Matrix Properties Trace

The trace of a matrix M is the sum of the diagonal elements of the matrix:

$$tr(M) = \sum_{i=1}^{n} M_{ij}$$

Definition 4.10: Matrix Properties ăInverse

The inverse matrix is a matrix that multiplies with the original matrix gives the identity:

$$MM^{-1} = I$$

Definition 4.11

Matrix Properties Rank The rank of a $n \times m$ matrix is the dimension of the space spanned by its columns.

The following properties for matrix derivation hold:

$$\frac{\partial M\boldsymbol{x}}{\partial \boldsymbol{x}} = M$$

$$\frac{\partial \boldsymbol{y}^T M\boldsymbol{x}}{\partial \boldsymbol{x}} = M^T\boldsymbol{y}$$

$$\frac{\partial \boldsymbol{x}^T M\boldsymbol{x}}{\partial \boldsymbol{x}} = \left(M^T + M\right)\boldsymbol{x}$$

$$\frac{\partial \boldsymbol{x}^T M\boldsymbol{x}}{\partial \boldsymbol{x}} = 2M\boldsymbol{x} \quad \text{if } M \text{ is symmetric}$$

$$\frac{\partial \boldsymbol{x}^T \boldsymbol{x}}{\partial \boldsymbol{x}} = 2\boldsymbol{x}$$

Definition 4.12: Norm

A function $\|\cdot\| : \mathcal{X} \to \mathbb{R}_0^+$ is a norm if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}, \lambda \in \mathbb{R}$:

- $\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$
- $\|\lambda\boldsymbol{x}\| = |\lambda| \, \|\boldsymbol{x}\|$
- $\|\boldsymbol{x}\| > 0$ if $\boldsymbol{x} \neq 0$

A norm defines a metric $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_0^+$:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|$$

Definition 4.13: Bilinear Form

A function $Q : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a bilinear form if for all $\cdot, \boldsymbol{y}.\boldsymbol{z} \in \mathcal{X}, \lambda, \mu \in \mathbb{R}$:

- $Q(\lambda \boldsymbol{x} + \mu \boldsymbol{y}, \boldsymbol{z}) = \lambda Q(\boldsymbol{x}, \boldsymbol{z}) + \mu Q(\boldsymbol{y}, \boldsymbol{z})$
- $Q(\boldsymbol{x}, \lambda \boldsymbol{y} + \mu \boldsymbol{z}) = \lambda Q(\boldsymbol{x}, \boldsymbol{y}) + \mu Q(\boldsymbol{x}, \boldsymbol{z})$

Definition 4.14: Bilinear Form Symmetry

A bilinear form is said to be *symmetric* if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$:

$$Q(\boldsymbol{x}, \boldsymbol{y}) = Q(\boldsymbol{y}, \boldsymbol{x})$$

Definition 4.15: Dot Product

A dot product $< \cdot, \cdot >: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a symmetric bilinear form which is positive semi-definite:

$$\langle \boldsymbol{x}, \boldsymbol{x} \rangle \geq 0 \forall \boldsymbol{x} \in \mathcal{X}$$

A positive definite dot product defines a corresponding norm via:

$$\|\boldsymbol{x}\| = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$$

One main property of the dot product is:

$$\cos\theta = \frac{\langle \boldsymbol{x}, \boldsymbol{z} \rangle}{\|\boldsymbol{x}\| \, \|\boldsymbol{z}\|}$$

From which follow that if two vectors are **orthogonal**, that is $\theta = \frac{\pi}{2} + k\pi, k \in \mathbb{N}$, then, $\langle \boldsymbol{x}, \boldsymbol{z} \rangle = 0$. Moreover a set of vectors $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ is said to be **orthonormal** if, for all vectors $\boldsymbol{x}_i, \boldsymbol{x}_j$ in the set:

$$\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

Definition 4.16: Eigenvalue and Eigenvector

Given an $n \times n$ matrix M, the real value λ and (non-zero) vector $\boldsymbol{x}$ are an eigenvalue and corresponding eigenvector of M if:

$$M\boldsymbol{x} = \lambda \boldsymbol{x}$$

Definition 4.17: Singular Matrix

A matrix is singular if it has a zero eigenvalue:

$$m\boldsymbol{x} = 0\boldsymbol{x} = 0$$

A singular matrix has linearly dependent columns:

$$\begin{bmatrix} M_1 & \dots & M_{n-1} & M_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{n_1} \\ x_n \end{bmatrix} = 0$$

$$M_1 x_1 + \dots + M_{n-1} x_{n-1} + M_n x_n = 0$$

$$M_n = M_1 \frac{-x_1}{x_n} + \dots + M_{n-1} \frac{-x_{n-1}}{x_n}$$

The determinant $|M|$ of a $n \times n$ matrix M is the product of its eigenvalues.
Since a matrix is invertible if its determinant is not zero, then it must not be singular.
We said that the eigenvalue and eigenvector of a matrix are a value and a vector such that:

$$A\boldsymbol{x} = \lambda \boldsymbol{x}$$

Trying to solve for λ we obtain:

$$\frac{\boldsymbol{x}^T A \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} = \lambda \frac{\boldsymbol{x}^T \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}$$

Definition 4.18: Raleigh quotient

$$\lambda = \frac{\boldsymbol{x}^T A \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}$$

5 Probability

5.1 Discrete

Definition 5.1: Probability Mass Function

Given a discrete random variable X taking values in $\mathcal{X} = \{v_1, \dots, v_m\}$, its probability mass function $P : \mathcal{X} \to [0, 1]$ is defined as:

$$P(v_i) = \Pr[X = v_i]$$

And satisfies the following conditions:

- $P(x) \geq 0$
- $\sum_{x \in \mathcal{X}} P(x) = 1$

So a variable is said to be discrete if it can assume m possible values $\{v_1, \dots, v_m\}$ which are mutual exclusive and if all probability of the values to happen are non-negative and the sum of probabilities of the events must be 1.

Definition 5.2: Expected Value

The expected value of a random variable x, also known as *mean* or *average*, is:

$$\mathrm{E}[x] = \mu = \sum_{x \in \mathcal{X}} xP(x) = \sum_{i=1}^{m} v_i P(v_i)$$

The expected value is linear:

$$\mathrm{E}[\lambda x + \lambda' y] = \lambda \mathrm{E}[x] + \lambda' \mathrm{E}[y] \tag{5.1}$$

Definition 5.3: Variance

The variance of a random variable is the moment of inertia of its probability mass function:

$$\mathrm{Var}[x] = \sigma^2 = \mathrm{E}[(x - \mu)^2] = \sum_{x \in \mathcal{X}} (x - \mu)^2 P(x)$$

Another important value is the standard deviation σ which indicates the typical amount of deviation from the mean.
The followings are properties of mean and variance:

- **Second moment**: it's similar to the expected value:

$$\mathrm{E}[x^2] = \sum_{x \in \mathcal{X}} x^2 P(x)$$

- It's possible to write the variance in term via the mean:

$$\mathrm{Var}[x] = \mathrm{E}[x^2] - \mathrm{E}[x]^2$$

- The variance is *not* always linear, for example when the variable is multiplied by a scalar:

$$\mathrm{Var}[\lambda x] = \lambda^2 \mathrm{Var}[x] \tag{5.2}$$

- The variance of the sum of two variables corresponds to the sum of the variance *only* if the two variables are not *correlated*:

$$\mathrm{Var}[x + y] = \mathrm{Var}[x] + \mathrm{Var}[y] \tag{5.3}$$

5.2 Discrete Probability Distribution

5.2.1 Bernoulli

This probabilistic distribution is used to model a binary event, that is an event that can only result in success or failure.
The probability p of an event x is the probability of success ($x = 1$), while the probability of failure ($x = 0$) is $1 - p$. The probability mass function is:

$$P(x, p) = \begin{cases} p & \text{if } x = 1 \\ 1 - p & \text{if } x = 0 \end{cases}$$

A scenarios that can be modelled using this distribution is for example the toss of a coin where head is success and tail is failure (or viceversa).

Theorem 5.1: Estimated Value – Bernoulli

$$\mathrm{E}[x] = p$$

Proof.

$$\mathrm{E}[x] = \sum_{x \in X} xP(x)$$
$$\mathrm{E}[x] = \sum_{x \in \{1,0\}} xP(x)$$
$$\mathrm{E}[x] = 0 \times (1 - p) + 1 \times p = p$$

□

Theorem 5.2: Variance – Bernoulli

$$\text{Var}[x] = p(1-p)$$

Proof.

$$\begin{aligned}
\text{Var}[x] &= \text{E}[(x-\mu)^2] = \\
&= \sum_{x \in X} (x - \text{E}[x])^2 P(x) = \\
&= \sum_{x \in X} (x-p)^2 P(x) = \\
&= (0-p)^2(1-p) + (1-p)^2 p = \\
&= p^2 - p^3 + (1 + p^2 - 2p)p = \\
&= p^3 - p^3 + p^2 + p - 2p^2 \\
&\quad \text{Var}[x] = p(1-p)
\end{aligned}$$

□

The Bernoulli probability can be expressed also via an analytic function which is often used in if-then-else cases:

$$p(x,p) = p^x (1-p)^{1-x}$$

If $x = 1$ then only the first term remains, resulting in $p(x,p) = p$, while if $x = 0$, then what remains is: $p(x,p) = 1 - p$

5.2.2 Binomial

Bernoulli distribution can be generalized with more than two events: this distribution models the probability of having a certain number of successes in n independent Bernoulli trials.
The parameter of this distribution are p has the probability of a success, and n as the number of trials.
The probability mass function is:

$$P(x; p, n) = \binom{n}{x} p^x (1-p)^{n-x}$$

Which represents the probability of success for x trials and can be seen as the Bernoulli distribution applied to all trials.
An example of event that can be modelled with this distribution is the toss of a coin for n times and trying to guess the probability of having x heads.
Mean and variation are:

$$\text{E}[x] = np \qquad \text{Var}[x] = np(1-p)$$

That is the same as the Bernoulli distribution but multiplied for the n tests.

5.2.3 Joint Probability

If two random variables are given instead of one, then the model must be different since they may not be independent.

Definition 5.4: Joint Probability

Given a pair of discrete random variables X, Y taking values $\mathcal{X} = \{v_1, \dots, v_m\}, \mathcal{Y} = \{w_1, \dots, w_n\}$, the joint probability mass function is defined as:

$$P(v_i, w_i) = \Pr[X = v_i, Y = w_j]$$

With properties:

- $P(x, y) \geq 0$
- $\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} P(x, y) = 1$

The characteristics of this function are:

- *Expected value*: since there are multiple random variables, then there will be one estimated value for each variable. Mind though that the mean for each variable is not simply the mean for that variable without considering the other variables, but is given by the sum of the value multiplied by the joint probability:

$$\mu_x = \mathrm{E}[x] = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} x P(x, y)$$

$$\mu_y = \mathrm{E}[y] = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} y P(x, y)$$

- *Variance*: as for the expected value, there will be one variance for each variable, and they will be computed based on the joint probability:

$$\sigma_x^2 = \mathrm{Var}[(x - \mu_x)^2] = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} (x - \mu_x)^2 P(x, y)$$

$$\sigma_y^2 = \mathrm{Var}[(y - \mu_y)^2] = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} (y - \mu_y)^2 P(x, y)$$

- *Covariance*: this is a statistics of how the random variables change at the change of another variable:

$$\sigma_{xy} = \mathrm{E}[(x - \mu_x)(y - \mu_y)] = \sum_{x \in \S} \sum_{y \in \mathcal{Y}} (x - \mu_x)(y - \mu_y) = P(x, y)$$

- *Correlation coefficient*: it is the normalization of the covariance with respect to the product of the variance:

$$\rho = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

5.2.3.1 Multinomial Distribution – One sample

Models the probability of a certain outcome for an event with $m > 2$ possible outcomes[1].
With a multinomial distribution, there are m parameters $p_1, \dots, p_m$ that indicate the probability

[1] If $m = 2$, then it's simply a Bernoulli distribution.

of the j outcome to happen.
For example this model represents the tossing of a dice one time, where m is the number of faces of the dice and p_i is the probability of face i to exit.
Since only one sample is considered, then the mass function of this distribution becomes the probability of that event to happen:

$$P(x_1, \dots, x_m; p_1 \dots, p_m) = \prod_{i=1}^{m} p_i^{x_i} \tag{5.4}$$

Where if $x_i = 1$, then all other outcomes must be 0: $x_j = 0, j \neq i$. The expected value, the variance are the same as the Bernoulli's while the covariance is:

$$\mathrm{E}[x_i] = p_i \qquad \mathrm{Var}[x_i] = p_i(1 - p_i) \qquad \mathrm{Cov}[x_i, x_j] = -p_i p_j$$

5.2.3.2 Multinomial Distribution – General

As for the Bernoulli distribution, also for the multinomial there exists a general version with more samples.
Given n samples of an event with m possible outcomes, the general version models the probability of a certain distribution of outcomes.
For example the following scenario is modelled by a general multinomial distribution: the toss of a coin with m faces, each with p_i probability, n times.
The mass function of this distribution is:

$$P(x_1, \dots, x_m; p_1 \dots, p_m) = \frac{n!}{\prod_{i=1}^{m} x_i!} \prod_{i=1}^{m} p_i^{x_i}$$

As for the Bernoulli distribution and the bionomial distribution, also in this case the statistics of the general distribution are the same as the one for one sample, just multiplied by n. The expected value, variance and covariance are:

$$\mathrm{E}[x_i] = np_i \qquad \mathrm{Var}[x_i = np_i(1 - p_i)] \qquad \mathrm{Cov}[x_i, x_j] = -np_i p_j$$

5.3 Conditional probabilities

In some cases we want to know the probability of an event to happen given that another event has happened. This is called *conditional probability*:

Definition 5.5: Conditional probability

The probability of x once y is observed:

$$P(x|y) = \frac{P(x, y)}{P(y)}$$

This implies the **product rule**:

Definition 5.6: Product rule

$$P(x, y) = P(x|y)P(y) = P(y|x)P(x)$$

This rule is useful because it allows to break down a joint probability into a combination of conditional probabilities.

Definition 5.7: Statistically independence

Two variable X and Y are said to be statistically independent if and only if:

$$P(x, y) = P(x)P(y) \tag{5.5}$$

This implies that if two variables X and Y are statistically independent, then:

$$P(x|y) = P(x) \qquad P(y|x) = P(y)$$

Definition 5.8: Low of Total Probability

The *marginal distribution* of a variable is obtained from a joint distribution summing over all possible values of the other variable (*sum rule*):

$$P(x) = \sum_{y \in \mathcal{Y}} P(x, y) \qquad P(y) = \sum_{x \in \mathcal{X}} P(x, y)$$

This means for example that considering a multinomial variable, it's possible to obtain its probability by removing it from the product and multiplying over the other values.

Definition 5.9: Bayes' Rule

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

Bayes' rule comes from the product rule and allows us to invert the dependency between two probabilities. In particular it allows to invert statistical connections between *effect*(x) and *cause*(y):

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

Mind that evidence can be obtained using the sum rule from likelihood and prior:

$$P(x) = \sum_{y} P(x, y) = \sum_{y} P(x|y)P(y)$$

These rules can be applied to more variables:

- **Sum rule**: $P(y) = \sum_{x} \sum_{z} P(x, y, z)$
- **Product rule**: apply the first rule more times, for example $P(x, y, z) = P(x|y, z)P(y, z) = P(x|y, z)P(y|z)P(z) = P(x, y, z) = P(x, y|z)p(z)$
- **Bayes' rule**: $P(y|x, z) = \dfrac{P(x|y, z)P(y|z)}{P(x|z}$

Let's compute the probability of y applying these rules:
For the sum rule (aka law of total), we know that:

$$P(y) = \sum_x \sum_z P(x, y, z)$$

And for the product rule we know that:

$$P(x, y, z) = P(y|x, z)P(x, z)$$

Hence:

$$P(y) = \sum_x \sum_z P(y|x, z)P(x, z)$$

At last applying Bayes' rule to $P(y|x, z)$ we obtain:

$$P(y) = \sum_x \sum_z \frac{P(x|y, z)P(y|z)P(x, z)}{P(x|z)}$$

5.3.1 Example of style

Let's start from the following expression and apply the rules:

$$P(y|x, z)$$

First of all let's apply Bayes' rule:

$$P(y|x, z) = \frac{P(x, z|y)P(y)}{P(x, z)}$$

Consider the discriminator $P(x, z)$, it's possible to apply the product rule:

$$P(y|x, z) = \frac{P(x, z|y)P(y)}{P(x|z)P(z)}$$

Now let's focus on the term $P(x, z|y)$. We can imagine it to come from $P(x, z, y)$ to which was applied the product rule: $P(x, z, y) = P(x, z|y)P(y)$, but it's also true that $P(x, z, y) = P(x|z, y)P(z|y)P(y)$, and hence $P(x, z|y) = P(x|z, y)P(z|y)$. It's possible then to rewrite the before as:

$$P(y|x, z) = \frac{P(x|z, y)P(z|y)P(y)}{P(x|z)P(z)}$$

Now focus on $P(z|y)P(y)$, it's obvious that by reversing the product rule, it equals $P(z, y)$, which can be written also as $P(y|z)P(z)$ since $P(z, y) = p(y, z)$:

$$P(y|x, z) = \frac{P(x|z, y)P(y|z)P(z)}{P(x|z)P(z)}$$

At last it's possible to simplify $P(z)$:

$$P(y|x, z) = \frac{P(x|z, y)P(y|z)}{P(x|z)}$$

5.4 Continuos variable

It's not possible to represent continuous variables with mass function since even if it had a small value, for as small it can be, we are summing it infinite times making it infinite.
Let's consider a continuous variable X and let's consider some intervals: $W = (a < X \leq b)$, $A = (x \leq a)$, $B(X \leq b)$. It's possible to notice that W and A are mutually esclusive, hence it's possible to write:

$$P(B) = P(A) + P(W) \qquad P(W) = P(B) - P(A)$$

Definition 5.10: Cumulative Distribution Function

A function $F(q)$ that models the probability of a continuous variable of having a value less or equal to q is called cumulative distributed function:

$$F(q) = P(X \leq q)$$

Mind that this is a monotonic function non decreasing: if the interval is enlarged, or it stays the same or it increases.
It's possible to get the probability of intervals by computing the difference between the cumulative distribution functions of the extremities:

$$P(a < X \leq) = F(b) - F(a)$$

Definition 5.11: Probability Density Function

The derivative of the cumulative distribution function is called probability density function and it represents the probability in a single point:

$$p(x) = \frac{d}{dx}F(x)$$

The cumulative distribution can also be computed from the density function integrating:

$$F(q) = P(X \leq q) = \int_{-\infty}^{q} p(x)dx$$

The following are properties of the density function:

- $p(x) \geq 0$: which implies that the density probability can also be greater than 1;
- $\int_{-\infty}^{\infty} p(x)dx = 1$: even if the density function can be grater than 1, the total integral over the possible values is 1.

Let's consider the following density distribution, the uniform distribution over $[a, b]$:

$$p(x) = Unif(x; a, b) = \frac{1}{b-a}$$

For $a = 0, b = {}^1/_2$, then $\forall x \in [0, {}^1/_2], p(x) = 2$. Let's now check the value of the integral:

$$F(0 < x \leq {}^1/_2) = \int_{-\infty}^{+\infty} p(x)dx = \int_a^b p(x)dx = \int_0^{\frac{1}{2}} 2dx = [2x]_0^{1/2} = 2 * \frac{1}{2} - 2 * 0 = 1$$

The expected value of a continuous variable, is computed as the integral of the product of the variable value by its probability:

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

Notice the similarity with the discrete value, only instead of the sum, an integral is used, as can be expected. It should come as no surprise then, the expression for the variance:

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

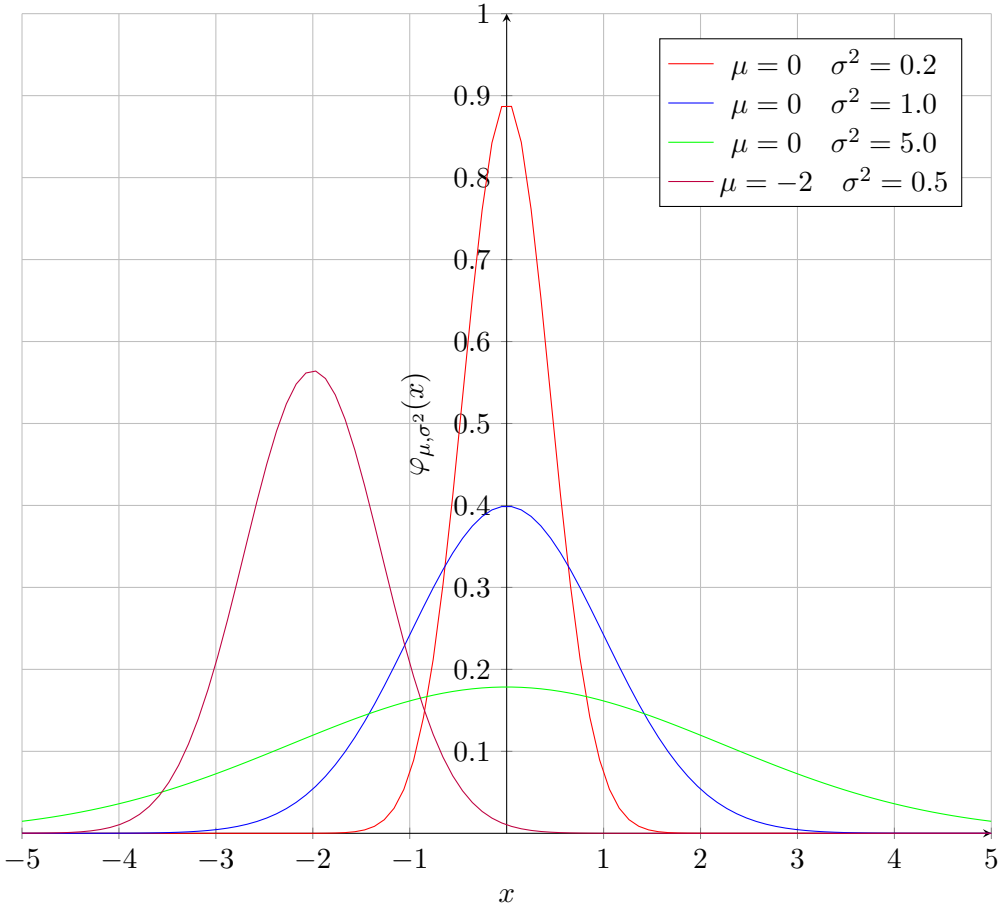
5.4.1 Gaussian – Normal

It is described in terms of μ and σ^2 which are its parameters.
It's density function is:

$$p(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} exp - \frac{(x-\mu)^2}{2\sigma^2} \tag{5.6}$$

Where the term $\frac{(x-\mu)^2}{2\sigma^2}$ is called standardisation of a normal distribution: $N(\mu,\sigma^2)$, and what it does is basically translate the belle shape and normalize it.
The standard normal distribution is a Gaussian distribution with mean 0 and variance 1.
As it can be expected, the expected value and the variance of the gaussian are μ and σ^2 respectively:

$$\mathrm{E}[x] = \mu \qquad \mathrm{Var}[x] = \sigma^2$$



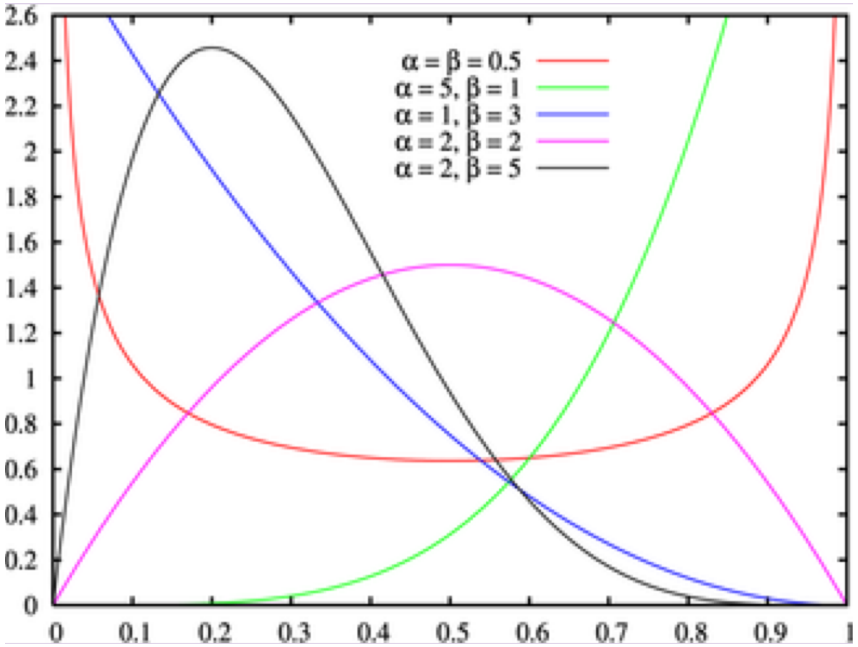
5.4.2 Beta Distribution

The Beta[2] distribution is defined inside the interval $[0, 1]$.
It is based on parameters α, β which let the density function vary as follows:

$$p(x; \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1} \tag{5.7}$$

This distribution can model for example the following scenario: we have a Bernoulli distribution, but no probability distribution for the event to happen is given. Such distribution can be modelled via a Beta distribution, making this a probability of the second order.
Notice that the terms $x^{\alpha-1}, (1-x)^{\beta-1}$ reminds of the binomial one, the beta distribution models the posterior distribution of parameter p of a binomial distribution after observing $\alpha - 1$ independent events with probability p and $\beta - 1$ events with probability $1 - p$.
The expected value and variation are:

$$\mathrm{E}[x] = \frac{\alpha}{\alpha + \beta} \qquad \mathrm{Var}[x] = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)} \tag{5.8}$$



5.4.3 Multivariate Normal Distribution

This is a normal distribution for d-dimensional vectorial data. For this reason the parameters are μ, a vector containing the means, Σ a covariance matrix.
It's possible to rewrite the gaussian distribution with the new data by exploiting matrices:

$$p(\boldsymbol{x}; \mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \tag{5.9}$$

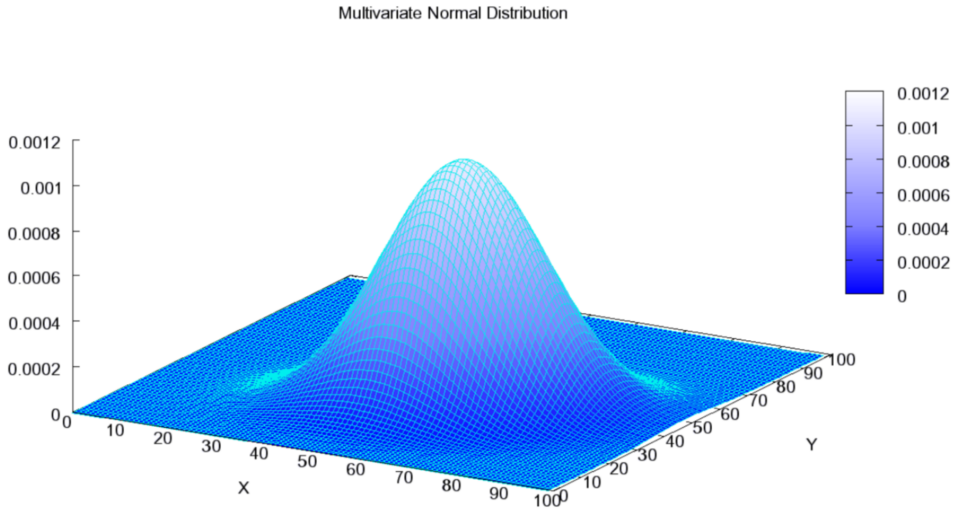
The term

$$\boldsymbol{r}^2 = (\boldsymbol{x} - \boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu}) \tag{5.10}$$

is called Mahalanobis distance and is the generalization of the euclidean distance with respect to the covariance of data. r^2 measures the distance from the mean $\boldsymbol{\mu}$.

[2]Pronounced /bit/.

The expected value is actually the expected one: $\mathrm{E}[x] = \mu$, while the variance is actually the covariance parameters: $\mathrm{Var}[x] = \Sigma$.



5.4.4 Dirichlet Distribution

The beta distribution is used to model for example the probability of a binary event. Obviously it can be generalized to the Dirichlet distribution, which is nothing less the continuous version of the multinomial distribution. The Dirichlet distribution allows to model the probability of an event with more than two possible results.

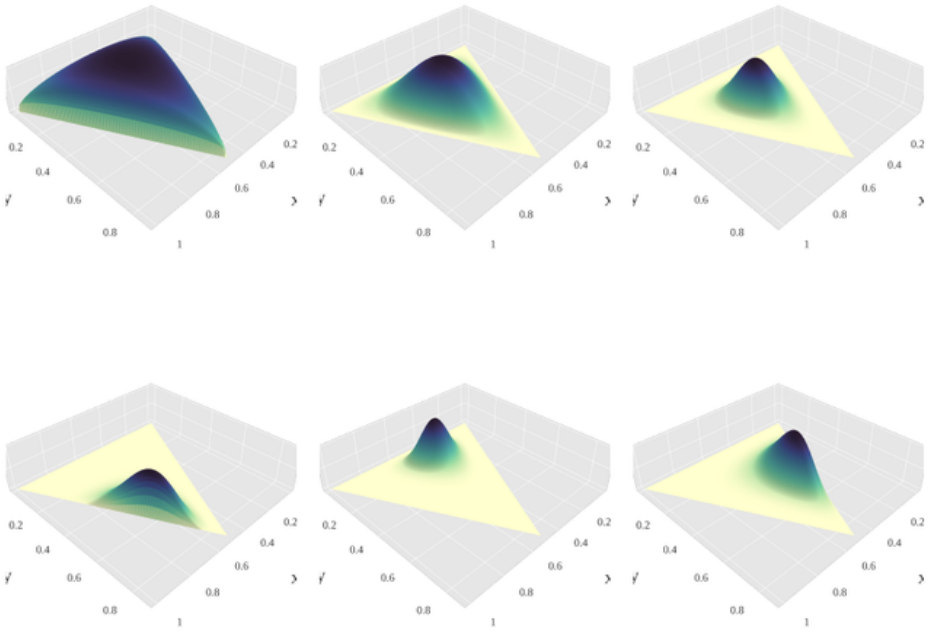
The Dirichlet distribution is defined for $\boldsymbol{x} \in [0,1]^m, \sum_{i=1}^{m} x_i = 1$, that is, given m possible outcomes, $x_1, \ldots, x_m$, the sum of all the outcomes must be 1.
It is parametrized over $\boldsymbol{\alpha} = \alpha_1, \ldots, \alpha_m$.
The probability density function is:

$$p(x_1, \ldots, x_m; \boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\prod_{i=1}^{m} \Gamma(\alpha_i)} \prod_{i=1}^{m} x_i^{\alpha_i - 1} \tag{5.11}$$

Where $\alpha_0 = \sum_{j=1}^{m} \alpha_j$. The expected value and variance are the followings:

$$\mathrm{E}[x_i] = \frac{\alpha_i}{\alpha_0} \qquad \mathrm{Var}[x_i] = \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)} \qquad \mathrm{Cov}[x_i, x_j] = \frac{-\alpha_i \alpha_j}{\alpha_0^2(\alpha_0 + 1)} \tag{5.12}$$

This distribution models the posterior distribution of parameters $\boldsymbol{p}$ of a multinomial distribution after observing $\alpha_i - 1$ times each mutually exclusive event.



5.5 Probability Laws

5.5.1 Expectation and Variance of an Average

Consider a sample of $X_1, \ldots, X_n$ *independent and identical distributed* (iid) instances drawn from a distribution with mean μ and variance σ^2.
Consider the random variable $\overline{X}_n$ measuring the sample average:

$$\overline{X}_n = \frac{X_1 + \ldots + X_n}{n}$$

Its expectation is computed as:

$$\mathrm{E}[\overline{X}_n] = \mathrm{E}[\frac{1}{n}(X_1 + \ldots + X_n)]$$

Now recall that the expected value is linear (Equation 5.1), hence it can be rewritten has:

$$\mathrm{E}[\overline{X}_n] = \frac{1}{n}(\mathrm{E}[X_1] + \ldots + \mathrm{E}[X_n])$$

Since $\mathrm{E}[X_i] = \mu$, then we have:

$$\mathrm{E}[\overline{X}_n] = \frac{1}{n}(\mu + \ldots + \mu) = \frac{1}{n} * n\mu = \mu$$

That is the expectation of an average is the true mean of the distribution.
Now let's consider the variance which though is not linear with respect to scalars (Equation **??**), while it behaves linear when summing variances (Equation 5.3). It's possible to write the variance of $\overline{X}_n$ as:

$$\mathrm{Var}[\overline{X}_n] = \frac{1}{n^2}(\mathrm{Var}[X_1] + \ldots + \mathrm{Var}[X_n]) = \frac{\sigma^2}{n}$$

From this is possible to notice that the variance of the average decreases with the number of observation: the more examples are taken into consideration, the more likely estimating the correct average is.

5.5.2 Chebyshev's Inequality

Consider a random variable X with mean μ and variance σ^2. The Chebyshev's inequality states that for all $a > 0$:

$$\Pr[|X - \mu| \geq a] \leq \frac{\sigma^2}{a^2}$$

Replacing a with $k\sigma, k > 0$ we obtain:

$$\Pr[|X - \mu| \geq k\sigma] \leq \frac{1}{k^2}$$

Chebyshev's inequality shows that most of the probability mass of a random variable stays within few standard deviations from its mean.

5.6 The Law of Large Numbers

Let's suppose to have some phenomenon to model and the distribution was not observed in its entirety, but only n events, for example we register only if rains for some days. Let's suppose also that the events are *identically distributed* since they come from the same phenomenon and that they are also independent which implies that the outcome of an event does not affect other events.
Our goal is to evaluate the average and the variance observing the data at our disposal.
Consider a sample of $X_1, \ldots, X_n$ iid instances drawn from a distribution with mean μ and variance σ^2.
We have already shown in the previous sections that the expected value of iid samples of mean μ is actually μ, now it's possible to show through the Chebyshev's inequality that the probability that the sample-mean drifts away from the actual mean of a little value
epsilon decreases at the increasing of the number of samples. This implies that the accuracy of an empirical statistic increases with the number of samples.

$$\Pr[|\overline{X}_n - \mathrm{E}[\overline{X}_n]| \geq \epsilon] \leq \frac{\sigma^2}{n\epsilon^2}$$

$$\lim_{n\to\infty} \Pr[|\overline{X}_n - \mu| \geq \epsilon] = 0$$

5.6.1 Central Limit Theorem

Definition 5.12: Central Limit Theorem

The sum of a sufficiently large sample of iid random measurement is approximately normally distributed.

This means that if we have a number n of samples sufficiently large, with mean μ and variance σ^2, we don't need to know the form of their distribution, because they can be modelled through a normal gaussian.
Questo significa che se abbiamo molti dati, se non conosciamo la distribuzione dei dati, allora la possiamo considerare questa distribuzione come una normale.

5.7 Information theory

5.7.1 Entropy

Consider a set of symbols $\mathcal{V} = \{v_1, \dots, v_n\}$ with mutually exclusive probabilities $P(v_i)$.
The goal is to design a binary code for each symbol minimizing the average length of messages.
In 1949 Shannon and Weaver proved that the optimal code assigns to each symbol v_i a number of bits equal to:

$$-\log P(v_i)$$

Definition 5.13: Entropy

The entropy of a set of symbols is the **expected value** length of a message encoding a symbol assuming such optimal coding:

$$H[\mathcal{V}] = \mathrm{E}[-\log P(v)] = -\sum_{i=1}^{n} P(v_i) \log P(v_i)$$

If all symbols have the same probability, then the entropy is maximized, while if the symbols have very much different probabilities, then the entropy becomes 0.

5.7.2 Cross entropy

Definition 5.14: Cross Entropy

Given two distributions P and Q over a variable X, the cross entropy between P and Q measures the expected number of bits needed to code a symbol sampled from P using Q instead:

$$H(P; Q) = \mathrm{E}_P[-\log Q(v)] = -\sum_{i=1}^{n} P(v_i) \log Q(v_i)$$

This is often used as a loss function for binary classification, with P (empirical) true distribution and Q (empirical) predicted distribution.

5.7.3 Relative Entropy

Definition 5.15: Relative Entropy

Given two distribution P, Q over a variable X, the relative entropy (or *Kullback-Leibler divergence)* measures the expected length difference when coding instances samples from P using Q instead:

$$D_{KL}(p \| q) = \sum_{i=1}^{n} P(v_i) \log \frac{P(v_i)}{Q(v_i)}$$

This can be derived as follows:

$$
\begin{aligned}
D_{KL}(p\|Q) &= H(P;Q) - H(P) \\
&= \mathrm{E}_P[-\log Q(v)] - \mathrm{E}[-\log P(v)] \\
&= -\sum_{i=1}^{n} P(v_i)\log Q(v_i) + \sum_{i=1}^{n} P(v_i)\log P(v_i) \\
&= \sum_{i=1}^{n} [P(v_i)\log P(v_i) - P(v_i)\log Q(v_i)] \\
&= \sum_{i=1}^{n} P(v_i)\left(\log P(v_i) - \log Q(v_i)\right) \\
&= \sum_{i=1}^{n} P(v_i)\log\frac{P(v_i)}{Q(v_i)}
\end{aligned}
$$

5.7.4 Conditional entropy

Definition 5.16: Conditional Entropy

Given two variables V, W with possibly different distributions P, the conditional entropy is the entropy remaining for variable W once V is known:

$$
\begin{aligned}
H(W|V) &= \sum_{v} P(v)H(W|V=v) \\
&= -\sum_{v} P(v)\sum_{w} P(w|v)\log P(w|v)
\end{aligned}
$$

This says that, the more we know about V, the more is the entropy of W, and viceversa.

5.7.5 Mutual Information – Information Gain

Definition 5.17: Information Gain

Given two variables V, W with (possibly different) distributions P, the mutual information, or information gain, is the reduction in entropy for W once V is known:

$$
\begin{aligned}
I(W;V) &= H(W) - H(W|V) \\
&= -\sum_{w} p(w)\log p(w) + \sum_{v} P(v)\sum_{w} P(w|v)\log P(w|v)
\end{aligned}
$$

The information gain is the reduction of entropy on W once V is known. This can be used, for example, when deciding which attribute is best to select when building a decision tree, where V is the attribute and W is the label.

6 Bayesian Decision Theory

We are not yet to the point of taking decision. Bayesian decision *theory* allows to take optimal decisions in a fully probabilistic way.
It allows to provide an upper bounds on achievable errors and evaluate classifiers accordingly. Moreover bayesian reasoning can be generalized to cases when the probabilistic structure is not entirely known.
From now on we will be using $P(x)$ for mass functions and $p(x)$ for distribution functions or unknowns.

6.1 Bayes decision Rule

Let's consider a binary classification. Assume of having examples $(x, y) \in \mathcal{X} \times \{-1, 1\}$ drawn from a known distribution $p(x, y)$. The task is predicting the class y of examples given the input x, which can be done via Bayes' rule (Definition 5.9):

$$P(y|x) = \frac{p(x|y)P(y)}{p(x)}$$

Bayes rule allows to compute the posterior probability given likelihood, prior and evidence:

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

Where:

- **posterior** $P(y|x)$ is the probability that the class is y given that x was observed;
- **Likelihood** $p(x|y)$ is the probability of observing x given that its class is y;
- **Prior**: $P(y)$: is the prior probability of the class, without any evidence;
- **Evidence**: $p(x)$ is the probability of the observation, and by the law of total probability -sum rule (Definition 5.8), and the product rule (Definition 5.6), it can be computed as:

$$p(x) = \sum_{i=1}^{2} p(x|y)P(y)$$

When taking decision it's possible to commit mistake, for this reason we want to know the probability of making mistakes.
The probability of an error can be expressed as the probability of the error given x on all possible values of y. For example when considering the binomial, then

$$P(error|x) = \begin{cases} P(y_2|x) & \text{if we decide } y_1 \\ P(y_1|x) & \text{if we decide } y_2 \end{cases}$$

For the sum rule (Definition 5.8), the average error can be expressed as:

$$P(error) = \sum_x P(error, x)$$

And by the product rule (Definition 5.6), it can be expanded to:

$$P(error) = \sum_x P(error|x)p(x)$$

This is for the discrete case, if the variable were to be continuous, then the formula would be:

$$P(error) = \int_{-\infty}^{\infty} P(error|x)p(x)dx$$

Based on the error probability value, a decision can be made in order to minimize the error maximize the probability of success. This rule is called Bayes Decision Rule and is divided in two cases; the first one is for binary classification:

$$y_B = \text{argmax}_{y_i \in \{-1,1\}} P(y_i|x) = \text{argmax}_{y_i \in \{-1,1\}} p(x|y_i)P(y_i) \tag{6.1}$$

Mind that by the Bayes Rule (Definition 5.9) the last expression should have $p(x)$ at the denominator, but this doesn't change with the changing of y_i, hence it can be removed.
A second case is the multiclass case:

$$y_B = \text{argmax}_{y_i \in \{1,\dots,c\}} P(y_i|x) = \text{argmax}_{y_i \in \{1,\dots,c\}} p(x|y_i)P(y_i) \tag{6.2}$$

The **optimal rule** says that the probability of error given x:

$$P(error|x) = 1 - P(y_B|x)$$

Hence, the Bayes Decision Rule *minimizes* the probability of error.

6.2 Representing Classifiers

So when we are making a decision, that is trying to classify an input, we are trying to maximize the probability a posteriori. To execute the classification, a *classifier* is used, that is a function, or more that we hope are as similar to the reality as they can be.
A classifier can be represented as a set of **discriminant functions** $g_i(\boldsymbol{x}), i\{\in 1, \dots, c\}$, where c is the number of classes. For such reason, the Bayes optimal rule can be rewritten as:

$$y = \text{argmax}_{i \in \{1,\dots,c\}} g_i(\boldsymbol{x})$$

By comparison with Equation 5.3, we have:

$$\begin{aligned} g_i(\boldsymbol{x}) = &\quad P(y_i|\boldsymbol{x} = \frac{p(\boldsymbol{x}|y_i)P(y_i)}{p(\boldsymbol{x})} \\ = &\quad p(\boldsymbol{x}|y_i)P(y_i) \end{aligned}$$

Which by the rule of logarithms can be rewritten as:

$$g_i(\boldsymbol{x}) = \ln p(\boldsymbol{x}|y_i) + \ln P(y_i)$$

With these discriminant functions, the features space is divided into decisions regions $\mathcal{R}_1, \dots, \mathcal{R}_c$ such that:

$$\boldsymbol{x} \in \mathcal{R}_i \quad \text{if } g_i(\boldsymbol{x}) > g_j(\boldsymbol{x}) \ \forall j \neq i$$

Decision regions are separated by **decision boundaries**, that is regions in which ties occur among the the discriminant functions and hence the classifier is ambiguous.
Let's suppose to have two features (ω_1 and ω_2) and to divide the space via two discriminant functions, hence binary. The most common choice to model likelihood is using gaussians which allows to have a multivariate normal distribution. From Figure 6.1 it's possible to notice the gaussians functions which has two peaks which represents the values for which the class will be chosen. For some values of x, the value of g will be the same, hence the decision boundaries. Let's first recall the equation for the multivariate normal density:

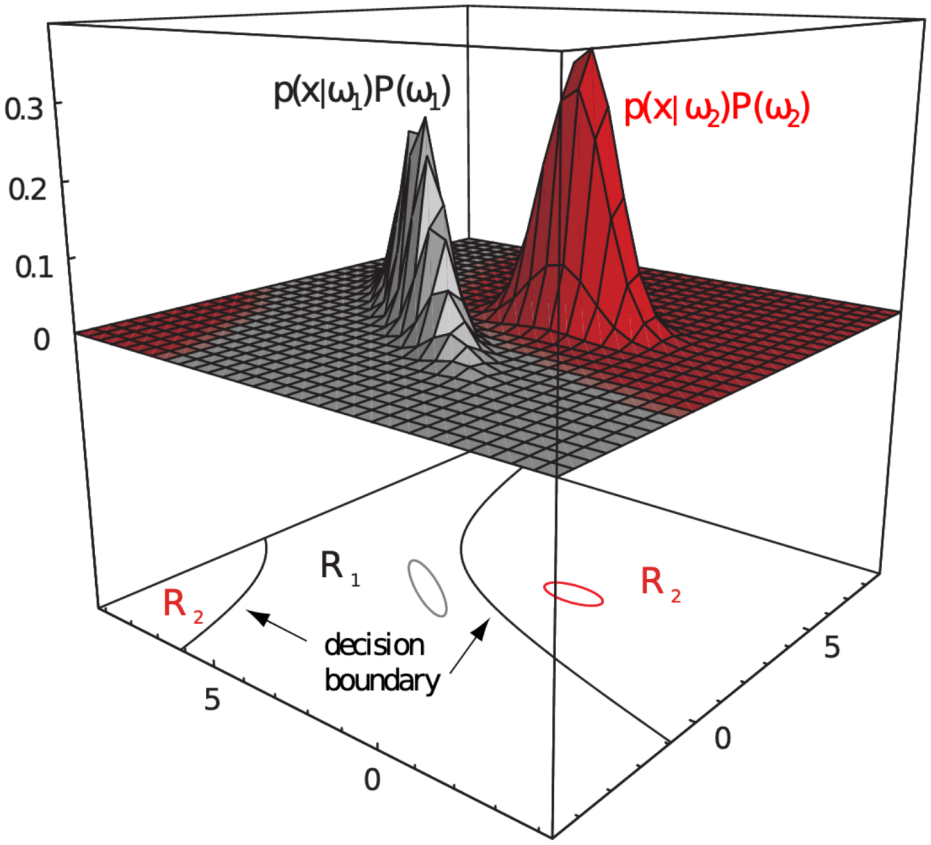


Figure 6.1: Example of classifiers.

$$p(\boldsymbol{x}; \mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$$

And let's remember that the covariance matrix Σ is always symmetric and positive semi-definite, and becomes strictly positive if the dimension of the feature space is f.
This distribution can map the probability of $\boldsymbol{x}$ given $\boldsymbol{y}_i$:

$$p(\boldsymbol{x}|y_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)}$$

Given a multivariate normal distribution, the location of points of constant density are hyper-ellipsoids of constant Mahalanobis distance (Equation 5.10) from $\boldsymbol{x}$ to $\boldsymbol{\mu}$.

6.2.1 Discriminant function

Let's know take expression from before describing discriminant functions:

$$g(x) = \ln P(x|y_i) + \ln P(y_i)$$

And substitute the multivariate normal distribution inside:

$$g_i(\boldsymbol{x}) = \ln \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)} + \ln P(y_i)$$

$$g_i(\boldsymbol{x}) = \ln \left[\frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}}\right] + \left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right] + \ln P(y_i)$$

$$g_i(\boldsymbol{x}) = -\frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_i| - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i) + \ln P(y_i)$$

Since $\frac{-}{d}2ln2\pi$ does not depend on i then we can erase it:

$$g_i(\boldsymbol{x}) = -\frac{1}{2}\ln|\Sigma_i| - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i) + \ln P(y_i) \tag{6.3}$$

6.2.1.1 $\Sigma_i = \sigma^2 I$

All features are statistically independent which means that they have the same variance σ^2.
Now the covariance determinant $|\Sigma_i| = \sigma^{2d}$ is independent of i and can hence be ignored from the Equation 6.3.
The inverse of the covariance is $\Sigma_i^{-1} = ({}^1/_{\sigma^2})I$ and even though is not dependent of i, it cannot be cancelled because part of another term. The new discriminant functions become:

$$g_i(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\frac{I}{\sigma^2}(\boldsymbol{x}-\boldsymbol{\mu}_i) + \ln P(y_i)$$

Which equals to:

$$g_i(\boldsymbol{x}) = -\frac{||\boldsymbol{x}-\boldsymbol{\mu}_i||^2}{2\sigma^2} + \ln P(y_i)$$

Expanding the quadratic term, we obtain:

$$g_i(\boldsymbol{x}) = -\frac{1}{2\sigma^2}\left[\boldsymbol{x}^T\boldsymbol{x} - 2\boldsymbol{\mu}_i^T\boldsymbol{x} + \boldsymbol{\mu}_i^T\boldsymbol{\mu}_i\right] + \ln P(y_i)$$

Discarding terms which are independent of i we obtain *linear discriminant functions*:

$$g_i(\boldsymbol{x}) = \underbrace{\frac{1}{\sigma^2}\boldsymbol{\mu}_i^T}_{\boldsymbol{w}_i^T}\boldsymbol{x} - \underbrace{-\frac{1}{2\sigma^2}\boldsymbol{\mu}_i^T\boldsymbol{\mu}_i + \ln P(y_i)}_{w_{i0}} \tag{6.4}$$

This is in fact linear: said $\boldsymbol{w}_i^T = {}^1/_{\sigma^2}\boldsymbol{\mu}_i^T\boldsymbol{x}$ and $w_{i0} = -{}^1/_{2\sigma^2}\boldsymbol{\mu}_i^T\boldsymbol{\mu}_i + \ln P(y_i)$, none of them depend on $\boldsymbol{x}$, hence $g_i(\boldsymbol{x})$ can be rewritten has:

$$g_i(\boldsymbol{x}) = \boldsymbol{w}_i^T\boldsymbol{x} + w_{i0}$$

which is indeed a line.
We have said that the boundaries are the region where the discriminant functions intersect, that is when they are equal $g_i(\boldsymbol{x}) = g_j(\boldsymbol{x})$. It's possible to find the equation of the decision boundaries:

$$g_i(\boldsymbol{x}) = g_j(\boldsymbol{x})$$
$$\frac{\boldsymbol{\mu}_i^T \boldsymbol{x}}{\sigma^2} - \frac{\boldsymbol{\mu}_i^T \boldsymbol{\mu}_i}{2\sigma^2} + \ln P(y_i) = \frac{\boldsymbol{\mu}_j^t \boldsymbol{x}}{\sigma^2} - \frac{\boldsymbol{\mu}_j^t \boldsymbol{\mu}_j}{2\sigma^2} + \ln P(y_j)$$
$$\underbrace{\frac{(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T}{\sigma^2}}_{\boldsymbol{w}^T} \boldsymbol{x} \underbrace{- \frac{\boldsymbol{\mu}_i \boldsymbol{\mu}_j}{\sigma^2} + \frac{\boldsymbol{\mu}_j^t \boldsymbol{\mu}_j}{\sigma^2} + \ln \frac{P(y_i)}{P(y_j)}}_{\boldsymbol{x}_0} = 0 \tag{6.5}$$

Which do not depend on $\boldsymbol{x}$ again hence making it linear again:

$$\boldsymbol{w}^T \boldsymbol{x} + \boldsymbol{x}_0$$

This line (which is an hyperplane btw) is orthogonal to the vector $\boldsymbol{w}$, that is to the distance between $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ since $\boldsymbol{w}$ is transposed with respect to $\boldsymbol{x}$.
Moreover the hyperplane passes through $\boldsymbol{x}_0$ which is based on the prior probabilities of classed being equal. If they are, then $\boldsymbol{x}_0$ is halfway between the means, while if they are not, $\boldsymbol{x}_0$ shifts away from the more likely mean. This can be seen in the logarithm: if $y_i = y_j$, then it would be $\ln 1 = 0$.

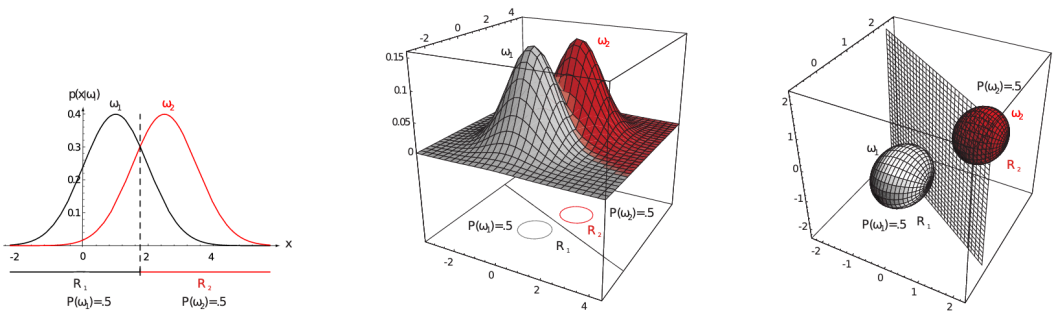


Figure 6.2: Case in which $P(y_i) = P(y_j)$. The discriminant function is a line which pass right through the middle of the distance of the two regions.

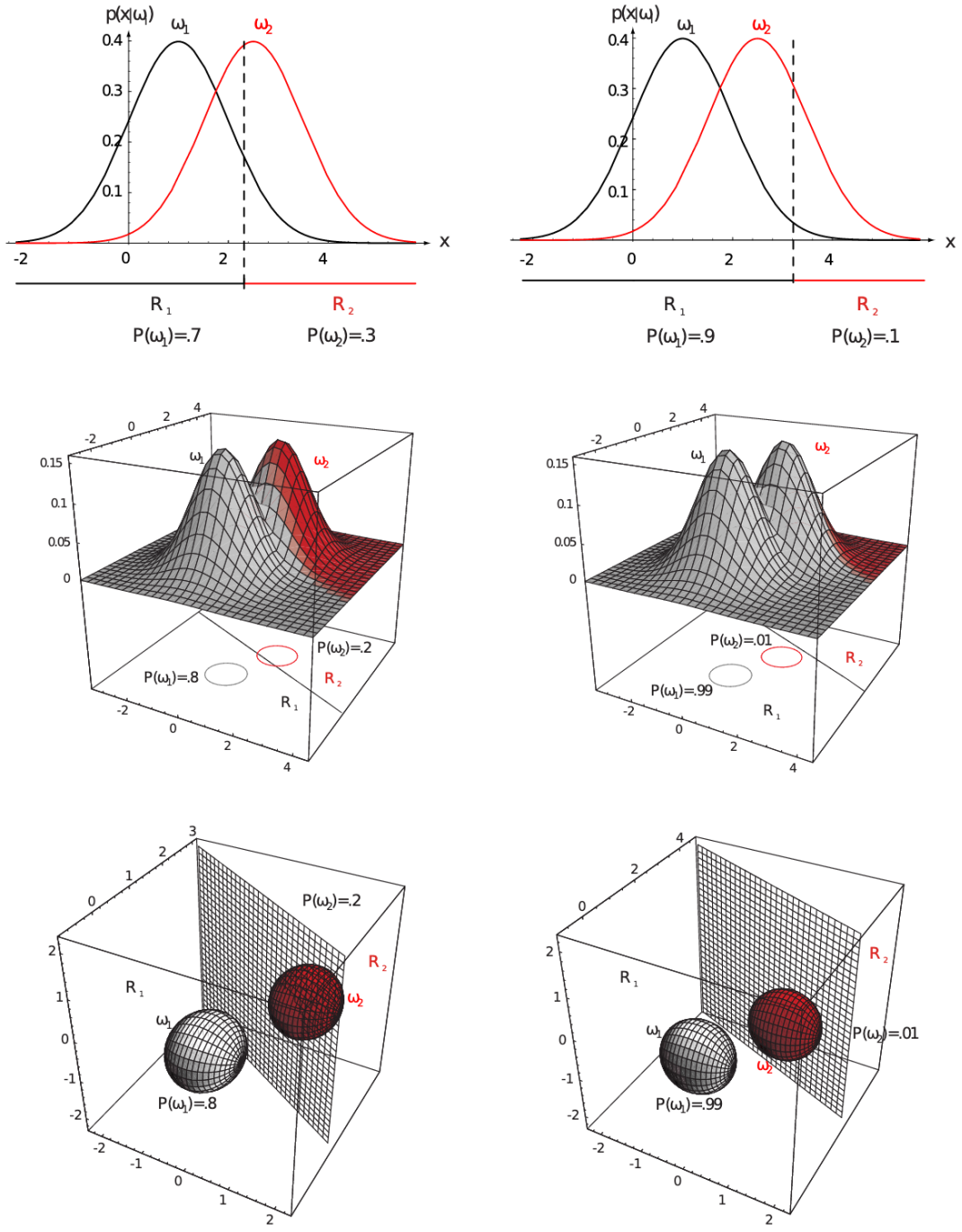


Figure 6.3: Case in which $P(y_i) \neq P(y_j)$. It's possible to notice that the more the probability difference increases, the more the line drifts away from that probability.

7 Maximum-Likelihood and Bayesian Parameter Estimation

Until now we have seen cases in which the parameters of the distributions were known. Now we'll look on how to estimate also this data.
Let's suppose to have data sampled from a probability distribution $p(x, y)$, but while the form of the distribution is known, the parameters of such are not. Let's suppose also to have a training set $\mathcal{D} = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ of examples sampled independently and identically distributed[1] according to $p(x, y)$. The goal is to estimate the unkown parameters of p from the training data $\mathcal{D}$.
Let's consider a multiclass classification problem: the training set can be divided into $\mathcal{D}_1, \ldots, \mathcal{D}_c$ subsets, one for each class. Mind that each subset contains iid examples for target class y_i: $\mathcal{D}_i = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$.
If the goal is to compute the probability of a class for a new example $\boldsymbol{x}$, then we can compute the probability of a class given the new example and all the dataset as described by Bayes rule, Definition 5.9:

$$P(y_i|\boldsymbol{x}, \mathcal{D}) = \frac{p(\boldsymbol{x}|y_i, \mathcal{D})p(y_i|\mathcal{D})}{p(\boldsymbol{x}|\mathcal{D})}$$

To really get the class of the new example we are going to compute the probability for all possible classes and then maximize over it.
Since $\boldsymbol{x}$ will be in a class y_i, it's safe to assume that it will be independent of $D_j, i \neq j$. It's possible to notice that the term $p(y_i|D)$ is the probability of finding a class inside the dataset, for example if the dataset contained students, and the class were to be Italian students, so it simply is:

$$p(y_i|\mathcal{D}) = \frac{|D_i|}{|D|}$$

As for the denominator, it's possible to marginalize (Sum rule 5.8) $p(\boldsymbol{x}|y_i, \mathcal{D}_i)p(y_i|\mathcal{D})$ over all possible classes:

$$p(\boldsymbol{x}|\mathcal{D}) = \sum_{y_i} p(\boldsymbol{x}|y_i, \mathcal{D}_i)p(y_i|\mathcal{D})$$

The idea of this step is that y_i does not depend on all $\mathcal{D}$, but only on $\mathcal{D}_i$.
The goal is to estimate *class-dependent* parameters θ_i for $p(\boldsymbol{x}|y_i, \mathcal{D}_i)$. There are two ways to solve this problem:

- **Bayesian estimation**: assumes that parameters θ_i are random variables with some known prior distribution. Once the example has been observed, the probability becomes a posterior distribution. Predictions for new examples are obtained integrating over all possible values for the parameters:

[1] *Independent*: each example is sampled independently from the others; *identically distributed*: all examples are sampled from the same distribution.

$$p(\boldsymbol{x}|y_i, \mathcal{D}_i) = \int_{\theta_i} p(\boldsymbol{x}, \theta_i|y_i, \mathcal{D}_i) d\theta_i$$

- **Maximum likelihood**: The parameters are computed as those maximising the probability of the observed examples $\mathcal{D}_i$, that is the training set for the class. For such reason they can be used to compute the probability for the new examples:

$$p(\boldsymbol{x}|y_i, \mathcal{D}_i) \approx p(\boldsymbol{x}|\theta_i)$$

7.1 Maximum likelihood

maximum likelihood Maximum likelihood If we can assume a prior distribution for the parameters $p(\theta_i)$, then maximum likelihood becomes maximum a-posteriori estimation and can be summarized as:

$$\theta_i^* = \text{argmax}_{\theta_i} p(\theta_i|\mathcal{D}_i, y_i) = \text{argmax}_{\theta_i} p(\mathcal{D}_i, y_i|\theta_i) p(\theta_i)$$

If a priori distribution is not available, then the solution is indeed maximum likelihood, that is maximizing the likelihood of the parameters with respect to the training samples:

$$\theta_i^* = \text{argmax}_{\theta_i} p(\mathcal{D}_i, y_i|\theta_i)$$

From now on, for simplicity, since each class y_i is trated independentely, we'll write $y_i, \mathcal{D}_\rangle$ as simply $\mathcal{D}$.
Given a training dataset $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ of iid examples for the target class y, assumed that the parameter vector $\boldsymbol{\theta}$ has a fixed but unknown value, we estimate such value by maximizing its likelihood with respect to the training data:

$$\theta^* = \text{argmax}_{\boldsymbol{\theta}} p(\mathcal{D}|\boldsymbol{\theta})$$

Since the examples are iid, the joint probability over $\mathcal{D}$ can be decomposed into a product:

$$\theta^* = \text{argmax}_{\boldsymbol{\theta}} \prod_{j=1}^{n} p(\boldsymbol{x}_j|\boldsymbol{\theta})$$

Moreover we don't generally like products, and since the goal is to maximize, it is possible to maximize over the logarithm of the product which equals to the sum of the logarithms. This is possible because the logarithm function is monotonic:

$$\theta^* = \text{argmax}_{\boldsymbol{\theta}} \ln p(\mathcal{D}|\boldsymbol{\theta}) = \text{argmax}_{\boldsymbol{\theta}} \ln \prod_{j=1}^{n} p(\boldsymbol{x}_j|\boldsymbol{\theta})$$

$$\theta^* = \text{argmax}_{\boldsymbol{\theta}} \sum_{j=1}^{n} \ln p(\boldsymbol{x}_j|\boldsymbol{\theta})$$

This is also called *maximizing the log-likelihood*.
The maxima can be obtained zeroing the gradient with respect to $\boldsymbol{\theta}$:

$$\nabla_{\boldsymbol{\theta}} = \sum_{j=1}^{n} \ln p(\boldsymbol{x}_j|\boldsymbol{\theta}) = \mathbf{0}$$

Since points zeroing the gradient can be local or global maxima.

7.1.1 Example – Univariate Gaussian

Let's consider a Gaussian distribution (Equation 5.6) with unknown μ and σ^2, then the log-likelihood $\mathcal{L}$ is:

$$\begin{aligned}\mathcal{L} = & \sum_{j=1}^{n} \ln \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x_j-\mu)^2}{2\sigma^2}} \\ = & \sum_{j=1}^{n} -\frac{1}{2\sigma^2}(x_j-\mu)^2 - \frac{1}{2}\ln 2\pi\sigma^2\end{aligned}$$

Now let's do the gradient with respect to θ, that is derive first for μ and then for σ^2:

$$\nabla_\theta = \begin{cases} \dfrac{\partial \mathcal{L}}{\partial \mu} \\ \dfrac{\partial \mathcal{L}}{\partial \sigma^2} \end{cases}$$

Let's do the one respect to μ first:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mu} = & -\sum_{j=1}^{n} \frac{1}{2\sigma^2}(2\mu - 2x_j) + 0 \\ = & \sum_{j=1}^{n} \frac{1}{\sigma^2}(x_j - \mu)\end{aligned}$$

Let's then do the one respect to σ^2:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \sigma^2} = & -\sum_{j=1}^{n} \frac{1}{2\sigma^2}(2\mu - 2x_j) - \frac{1}{2}\ln 2\pi\sigma^2 \\ = & -\sum_{j=1}^{n} \left[\frac{1}{2}(x_j-\mu)^2 \frac{\partial}{\partial\sigma^2}\frac{1}{\sigma^2}\right] - \frac{n}{2}\frac{\partial}{\partial\sigma^2}\ln 2\pi\sigma^2 \\ = & -\frac{n}{2}2\pi\frac{1}{2\pi\sigma^2} - \sum_{j=1}^{n}\frac{(x_j-\mu)^2}{2}(-1)\frac{1}{\sigma^4} \\ = & -\frac{n}{2\sigma^2} + \sum_{j=1}^{n}\frac{(x_j-\mu)^2}{2\sigma^4}\end{aligned}$$

The gradient with respect to the parameters is:

$$\nabla_\theta = \begin{cases} \dfrac{\partial \mathcal{L}}{\partial \mu} = \displaystyle\sum_{j=1}^{n} \frac{1}{\sigma^2}(x_j - \mu) \\ \dfrac{\partial \mathcal{L}}{\partial \sigma^2} = -\dfrac{n}{2\sigma^2} + \displaystyle\sum_{j=1}^{n} \frac{(x_j-\mu)^2}{2\sigma^4} \end{cases}$$

Now we need to zero the gradient. First for μ:

$$\begin{aligned}\sum_{j=1}^{n} \frac{1}{\sigma^2}(x_j - \mu) = 0 \\ \sum_{j=1}^{n}(x_j - \mu) = 0\end{aligned}$$

$$\sum_{j=1}^{n} x_j - \sum_{j=1}^{n} \mu = 0$$
$$\sum_{j=1}^{n} x_j = \sum_{j=1}^{n} \mu$$
$$\sum_{j=1}^{n} x_j = n\mu$$
$$\mu = \frac{1}{n}\sum_{j=1}^{n} x_j$$

Now let's do the one for σ^2:

$$-\frac{n}{2\sigma^2} + \sum_{j=1}^{n} \frac{(x_j - \mu)^2}{2\sigma^4} = 0$$
$$\sum_{j=1}^{n} \frac{(x_j - \mu)^2}{2\sigma^4} = \frac{n}{2\sigma^2}$$
$$\frac{1}{2\sigma^4}\sum_{j=1}^{n}(x_j - \mu)^2 = \frac{n}{2\sigma^2}$$
$$\frac{1}{\sigma^2}\sum_{j=1}^{n}(x_j - \mu)^2 = n$$
$$\sigma^2 = \frac{1}{n}\sum_{j=1}^{n}(x_j - \mu)^2$$

So the maximum-likelihood estimates are:

$$\mu = \frac{1}{n}\sum_{j=1}^{n} x_j$$

$$\sigma^2 = \frac{1}{n}\sum_{j=1}^{n}(x_j - \mu)^2$$

7.1.2 Example – Multivariate Gaussian

Let's recall the multivariate distribution (Equation 5.9):

$$p(\boldsymbol{x}; \mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$$

The log-likelihood is:

$$\sum_{j=1}^{n} -\frac{1}{2}(\boldsymbol{x}_j - \boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{x}_j - \boldsymbol{\mu}) - \frac{1}{2}\ln(2\pi)^d |\Sigma|$$

And the maximum -likelihood estimates are:

$$\mu = \frac{1}{n}\sum_{j=1}^{n} \boldsymbol{x}_j$$
$$\Sigma = \frac{1}{n}\sum_{j=1}^{n} (\boldsymbol{x}_j \boldsymbol{\mu})(\boldsymbol{x}_j - \boldsymbol{\mu})^T$$

It's possible to notice then, that for Gaussian distributions, the parameters are simply their empirical estimates over the sample:

- Gaussian mean is the sample mean;
- Gaussian covariance matrix is the mean of the sample covariances.

7.2 Bayesian estimation

Let's assume that parameters θ_i are *random variables* with some known *prior* distribution.
Predictions for new examples are obtained integrating over all possible values for the parameters:

$$p(\boldsymbol{x}|y_i, \mathcal{D}_i) = \int_{\boldsymbol{\theta}_i} p(\boldsymbol{x}, \boldsymbol{\theta}_i|y_i, \mathcal{D}_i) d\boldsymbol{\theta}_i$$

Probability of $\boldsymbol{x}$ given each class y_i is independent of the other classes y_i, for simplicity we can again write:

$$p(\boldsymbol{x}|y_i, \mathcal{D}_i) \rightarrow p(\boldsymbol{x}|\mathcal{D}) = \int_{\boldsymbol{\theta}} p(\boldsymbol{x}, \boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta}$$

Where $\mathcal{D}$ is a dataset for a certain class y, and $\boldsymbol{\theta}$ the parameters of the distribution.

$$p(\boldsymbol{x}|\mathcal{D}) = \int_{\boldsymbol{\theta}} p(\boldsymbol{x}, \boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta} = \int_{\boldsymbol{\theta}} p(x|\boldsymbol{\theta}, D) p(\boldsymbol{\theta}|D) d\boldsymbol{\theta} = \int_{\boldsymbol{\theta}} p(\boldsymbol{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta} \tag{7.1}$$

Where $p(\boldsymbol{x}|\mathcal{D})$ can be easily computed since we have both form and parameters of the distribution, e.g., if it were a gaussian the form is Equation 5.6, while the parameters are μ and σ^2.
Since our goal is to estimate the parameters' posterior density given a training set $p(\boldsymbol{\theta}|\mathcal{D})$ by using Bayes rule (Definition 5.9), then we can write:

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{D})} = \frac{\left(\prod_{j=i}^{n} (P(\boldsymbol{x}_i|\boldsymbol{\theta}) \right) P(\boldsymbol{\theta})}{P(\mathcal{D})}$$

$P(\mathcal{D}|\boldsymbol{\theta})$ is the likelihood probability that has been used also in the maximum likelihood method. In this case though, the parameters are weighted with respect to their probability avoiding overfitting which is a problem of maximum likelihood.
Mind that $P(\mathcal{D})$ is a constant independent of θ, hence it will have no influence in the final Bayesian decision, therefore it can be neglected.
At last, if final probability is needed, then $P(\mathcal{D})$ is computable via:

$$p(\mathcal{D}) = \int_{\boldsymbol{\theta}} p(\mathcal{D}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}$$

7.2.1 Example – Univariate Normal, unknown μ, known σ^2

Example are drawn from a normal distribution, hence:

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = p(x|\mu) \sim N(\mu, \sigma^2)$$

Moreover the Gaussian mean distribution is itself a normal distribution:

$$P(\mu) \sim N(\mu_0, \sigma_0^2)$$

So the distribution of the parameter's posterior distribution can be computed as:

$$p(\mu|\mathcal{D}) = \frac{p(\mathcal{D}|\mu)p(\mu)}{p(\mathcal{D})} = \frac{1}{p(\mathcal{D})}\left(\prod_{j=1}^{n} p(x_j|\mu)\right)p(\mu)$$

Where we said that $p(x_j|\mu) \sim N(\mu, \sigma^2)$ and $p(\mu) \sim N(\mu_0, \sigma_0^2)$:

$$p(\mu|\mathcal{D}) = \frac{1}{p(\mathcal{D})}\left(\prod_{j=1}^{n} \underbrace{\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_j-\mu)^2}{2\sigma^2}}}_{p(x_j|\mu)}\right) \underbrace{\frac{1}{\sigma_0\sqrt{2\pi}} e^{-\frac{(\mu-\mu_0)^2}{2\sigma_0^2}}}_{p(\mu)}$$

Since $p(\mathcal{D})$ is constant, it can be moved out of the product, and we'll call it $\alpha = {}^{1}/_{p(\mathcal{D})}$:

$$p(\mu|\mathcal{D}) = \alpha \prod_{j=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_j-\mu)^2}{2\sigma^2}} \frac{1}{\sigma_0\sqrt{2\pi}} e^{-\frac{(\mu-\mu_0)^2}{2\sigma_0^2}}$$

Now we could use the exponentiation's properties:

$$e^x \times e^y = e^{x+y}$$

$$\prod_i e^{x_i} = e^{\Sigma_i x_i}$$

Before though, the constant parts could be moved outside the product and we'll call the new constant part outside the product as α':

$$p(\mu|\mathcal{D}) = \alpha \left(\frac{1}{2\sigma\sigma_0\pi}\right)^n \left(\prod_{j=1}^{n} \exp\left\{-\frac{(x_j-\mu)^2}{2\sigma^2}\right\}\right) \exp\left\{-\frac{(\mu-\mu_0)^2}{2\sigma_0^2}\right\}$$

$$p(\mu|\mathcal{D}) = \alpha' \exp\left\{\sum_{j=1}^{n} -\frac{(x_j-\mu)^2}{2\sigma^2}\right\} \exp\left\{-\frac{(\mu-\mu_0)^2}{2\sigma_0^2}\right\}$$

$$p(\mu|\mathcal{D}) = \alpha' \exp\left\{\frac{1}{2}\left(\sum_{j=1}^{n} \frac{(x_j-\mu)^2}{\sigma^2} + \frac{(\mu-\mu_0)^2}{\sigma_0^2}\right)\right\}$$

Let's now focus only on the exponent[2]:

$$-\frac{1}{2}\left(\sum_{j=1}^{n} \frac{(\mu-x_j)^2}{\sigma^2} + \frac{(\mu-\mu_0)^2}{\sigma_0^2}\right)$$

Now it's possible to expand the exponentials:

$$-\frac{1}{2}\left(\sum_{j=1}^{n} \frac{\mu^2 + x_j^2 - 2\mu x_j}{\sigma^2} + \frac{\mu^2 + \mu_0^2 - 2\mu\mu_0}{\sigma_0^2}\right)$$

And move the constant term wrt to the sum outside the sum:

$$-\frac{1}{2}\left(\frac{n\mu^2}{\sigma^2} + \frac{1}{\sigma^2}\sum_{j=1}^{n} x_j^2 - \frac{1}{\sigma^2}\sum_{j=1}^{n} 2\mu x_j + \frac{\mu^2}{\sigma^2} + \frac{\mu_0^2}{\sigma^2} - \frac{2\mu\mu_0}{\sigma_0^2}\right)$$

[2] $(x_j - \mu)^2 = (\mu - x_j)^2$.

First of all let's group the common terms:

$$-\frac{1}{2}\left(\mu^2\left(\frac{n}{\sigma^2}+\frac{1}{\sigma^2}\right)+\frac{\mu_0^2}{\sigma^2}+\frac{1}{\sigma^2}\sum_{j=1}^{n}x_j^2-2\mu\left(\frac{1}{\sigma^2}\sum_{j=1}^{n}x_j-\frac{\mu_0}{\sigma_0^2}\right)\right)$$

Let's now consider all the equation obtained:

$$p(\mu|\mathcal{D})=\alpha'\exp\left\{-\frac{1}{2}\left(\mu^2\left(\frac{n}{\sigma^2}+\frac{1}{\sigma^2}\right)+\frac{\mu_0^2}{\sigma^2}+\frac{1}{\sigma^2}\sum_{j=1}^{n}x_j^2-2\mu\left(\frac{1}{\sigma^2}\sum_{j=1}^{n}x_j-\frac{\mu_0}{\sigma_0^2}\right)\right)\right\}$$

Mind now that this is, by the exponentials properties, the same as:

$$p(\mu|\mathcal{D})=\alpha'\exp\left\{-\frac{1}{2}\left(\mu^2\left(\frac{n}{\sigma^2}+\frac{1}{\sigma^2}\right)-2\mu\left(\frac{1}{\sigma^2}\sum_{j=1}^{n}x_j-\frac{\mu_0}{\sigma_0^2}\right)\right)\right\}\exp\left\{+\frac{\mu_0^2}{\sigma^2}+\frac{1}{\sigma^2}\sum_{j=1}^{n}x_j^2\right\}$$

Where the second exponential is actually not dependent on μ, hence it can be moved insider α' which then becomes $\alpha''=\alpha'\times\exp\left\{+\frac{\mu_0^2}{\sigma^2}+\frac{1}{\sigma^2}\sum_{j=1}^{n}x_j^2\right\}$:

$$p(\mu|\mathcal{D})=\alpha''\exp\left\{-\frac{1}{2}\left(\mu^2\left(\frac{n}{\sigma^2}+\frac{1}{\sigma^2}\right)-2\mu\left(\frac{1}{\sigma^2}\sum_{j=1}^{n}x_j-\frac{\mu_0}{\sigma_0^2}\right)\right)\right\}$$

It's possible to notice that this posterior probability is quite similar with a normal, let's describe it as:

$$p(\mu|\mathcal{D})=\frac{1}{\sqrt{2\pi}\sigma_n}\exp\left\{-\frac{1}{2}\left(\frac{\mu-\mu_n}{\sigma_n}\right)^2\right\} \tag{7.2}$$

$$p(\mu|\mathcal{D})=\frac{1}{\sqrt{2\pi}\sigma_n}\exp\left\{-\frac{1}{2}\left(\frac{\mu^2}{\sigma_n^2}+\frac{\mu_n^2}{\sigma_n^2}-\frac{2\mu\mu_n}{\sigma_n^2}\right)\right\}$$

$$p(\mu|\mathcal{D})=\frac{1}{\sqrt{2\pi}\sigma_n}\exp\left\{\frac{\mu_n^2}{\sigma_n^2}\right\}\exp\left\{-\frac{1}{2}\left(\frac{\mu^2}{\sigma_n^2}-\frac{2\mu\mu_n}{\sigma_n^2}\right)\right\}=\alpha'''\exp\left\{-\frac{1}{2}\left(\frac{\mu^2}{\sigma_n^2}-\frac{2\mu\mu_n}{\sigma_n^2}\right)\right\}$$

By comparison with the equation obtained before it's possible to notice that:

$$\frac{1}{\sigma_n^2}=\frac{n}{\sigma^2}+\frac{1}{\sigma_0^2}\qquad\frac{\mu_n}{\sigma_n^2}=\frac{\mu_0}{\sigma_0^2}+\sum_{j=1}^{n}x_j$$

Let's first focus on the second equation. By multiplying and dividing for n the sum, it's possible to obtain the sample mean $\widehat{\mu}_n$:

$$\frac{\mu_n}{\sigma_n^2}=\frac{\mu_0}{\sigma_0^2}+n\frac{1}{n}\sum_{j=1}^{n}x_j=\frac{\mu_0}{\sigma_0^2}+n\widehat{\mu}_n$$

Now let's solve for σ_n first since μ_n depends on σ_n:

$$\frac{1}{\sigma_n^2}=\frac{n\sigma_0^2+\sigma^2}{\sigma^2\sigma_0^2}$$

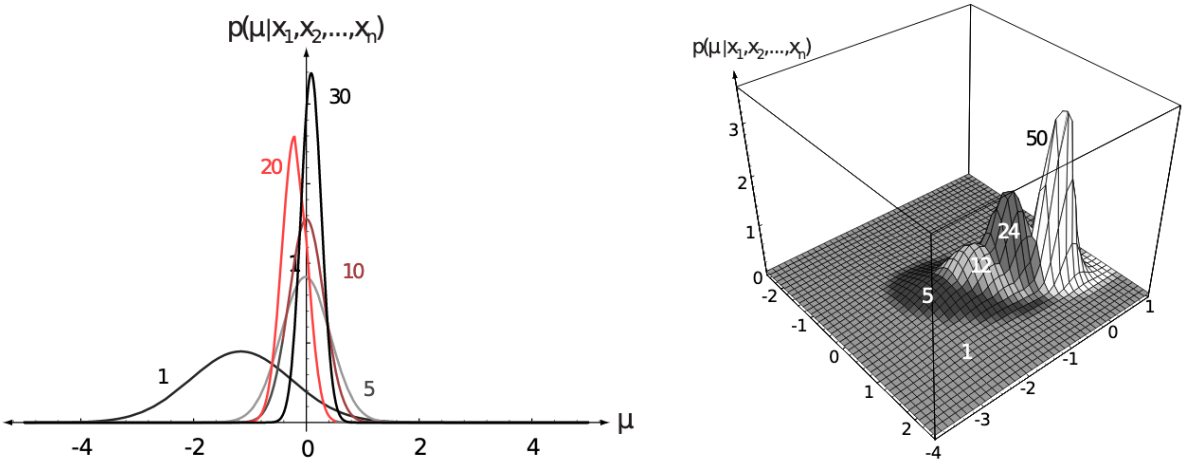
$$\sigma_n^2=\frac{\sigma^2\sigma_0^2}{n\sigma_0^2+\sigma^2}$$

It's now possible to solve for μ_n:

$$\mu_n = n\widehat{\mu}_n\sigma_n^2 + \frac{\mu_0\sigma_n^2}{\sigma_0}$$

$$\mu_n = n\widehat{\mu}_n\frac{\sigma^2\sigma_0^2}{n\sigma_0^2+\sigma^2} + \mu_0\frac{\sigma^2}{n\sigma_0^2+\sigma^2}$$

Notice that the men μ_n is a linear combination of the prior μ_0 and the sample mean $\widehat{\mu}_n$. If the number of samples increases, the sample mean starts to dominate over the prior mean and the variance decreases, making the distribution sharply peaked over its mean.



We stil haven't computed the class conditional density $p(x|\mathcal{D})$. Remember Equation 7.1:

$$P(x|\mathcal{D}) = \int_\mu P(x|\mu)P(\mu|\mathcal{D})\mu$$

Since we said that $p(x|\mu) \sim N(\mu, \sigma^2)$, and from what was achieved in Equation 7.2, it's possible to write:

$$p(x|\mathcal{D}) = \int_\mu N(\mu,\sigma^2)N(\mu_n,\sigma_n^2)d\mu$$

$$p(x|\mathcal{D}) = \int_\mu \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right\}\frac{1}{\sqrt{2\pi}\sigma_n}\exp\left\{-\frac{1}{2}\left(\frac{\mu-\mu_n}{\sigma_n}\right)^2\right\}d\mu$$

Let's now define $f(\sigma,\sigma_n)$:

$$f(\sigma,\sigma_n) = \int_\mu \exp\left\{-\frac{1}{2}\frac{\sigma^2+\sigma_n^2}{\sigma^2\sigma^n}\left(\mu - \frac{\sigma_n^2 x + \sigma^2\mu_n}{\sigma^2+\sigma_n^2}\right)^2\right\}d\mu$$

Then we could rewrite the previous probability as:

$$p(x|\mathcal{D}) = \int_\mu \frac{1}{2\pi\sigma\sigma_n}\underbrace{\exp\left\{-\frac{1}{2}\frac{(x-\mu_n)^2}{\sigma^2+\sigma_n^2}\right\}}_{\beta} f(\sigma,\sigma_n)$$

If we consider $p(x|\mathcal{D})$ as a function of x, then it is proportional to β in the previous equation and hence $p(x|\mathcal{D})$ is normally distributed with mean μ_n and variance $\sigma^2+\sigma_n^2$:

$$p(x|\mathcal{D}) \sim N(\mu_n, \sigma^2+\sigma_n^2)$$

This means that the probability of x given the dataset for the class is a Gaussian with mean equal to the posterior mean, and the variance equal to the sum of the known variance (σ^2) and the additional variance (σ^2)due to the uncertainty of the mean.
The same thing happens with the multivariate. Known that the examples are drawn from a multivariate and that also distribution of the mean for the multivariate is still a multivariate:

$$p(\boldsymbol{x}|\boldsymbol{\mu}) \sim N(\boldsymbol{\mu}, \Sigma)$$

$$p(\boldsymbol{\mu}) \sim N(\boldsymbol{\mu}_0, \Sigma_0)$$

It's possible to obtain the posterior and the class-conditional distributions:

$$p(\boldsymbol{\mu}|\mathcal{D}) \sim N(\boldsymbol{\mu}_n, \Sigma_n)$$

$$p(\boldsymbol{x}|\mathcal{D}) \sim N(\boldsymbol{\mu}_n, \Sigma + \Sigma_n)$$

7.3 Sufficient statics

Definition 7.1: Statistic

A function on a set of samples $\mathcal{D}$ is called statistic.

One example of statistics is the mean. We'll use statistics to evaluate the forecast.

Definition 7.2: Sufficient Statistic

A statistic $\phi(\mathcal{D})$is said to be *sufficient* for some parameters $\boldsymbol{\theta}$ if all we need to estimate the parameter is the statistic:

$$P(\mathcal{D}|\boldsymbol{s}, \boldsymbol{\theta}) = P(\mathcal{D}|\boldsymbol{s})$$

If $\boldsymbol{\theta}$ is a random variable, a sufficient statistic contains all relevant information $\mathcal{D}$ has for estimating it:

$$p(\boldsymbol{\theta}|\mathcal{D},) = \frac{p(\mathcal{D}|\theta, \boldsymbol{s})p(\boldsymbol{\theta}|\boldsymbol{s})}{p(\mathcal{D}|\boldsymbol{S})} = p(\boldsymbol{\theta}|\boldsymbol{s})$$

For example if we consider the Gaussian, the sample mean was all one would need from the dataset, the exact values are meaningless.

8 10/10/2019

8.1 Conjugate

Definition 8.1: Conjugate priors

Given a likelihood function $p(x|\theta)$, given a prior distribution $p(\theta)$, $p(\theta)$ is said to be a conjugate prior for $p(x|\theta)$ if the posterior distribution $p(\theta|x)$ is the same family as the prior $p(\theta)$.

Over what was seen before about the univariate normal case, other examples are described in Table 8.1. For example if we are trying to estimate samples taken from a Normal distribution ($p(x\boldsymbol{\theta}) \sim N(\mu, \sigma^2)$), and the modelled parameter is the mean, then the conjugate prior will still be a normal distribution.

Likelihood	Parameters	Conjugate Prior
Binomial	p (probability)	Beta
Multinomial	$\boldsymbol{p}$ (probability vector)	Dirichlet
Normal	μ (mean)	Normal
Multivariate Normal	$\boldsymbol{\mu}$ (mean vector)	Normal

Table 8.1: Table showing the conjugate prior of a likelihood depending on the parameter.

8.1.1 Bernoulli Distribution

This distribution models boolean events, either they are "success", or they are "failure". The parameter θ of the Bernoulli is simply the probability of success. The mass function (which will be used in the likelihood) is:

$$P(x|\theta) = \theta^x(1-\theta)^{1-x}$$

The conjugate prior is a Beta distribution:

$$P(\theta|\psi) = P(\theta|\alpha_h, \alpha_t) = \frac{\Gamma(\alpha)}{\Gamma(\alpha_h)\Gamma(\alpha_t)}\theta^{\alpha_h - 1}(1-\theta)^{\alpha_t - 1}$$

8.1.1.1 Example

Let's consider a dataset made out of the results of tossing a coin:

$$\mathcal{D} = \{H, H, T, T, T, H, H\}$$

The parameter θ indicates the probability of tossing head. The likelihood function becomes:

$$p(\mathcal{D}|\theta) = \theta \cdot \theta \cdot (1-\theta) \cdot (1-\theta) \cdot (1-\theta) \cdot \theta \cdot \theta = \theta^h(1-\theta)^t$$

First let's try to estimate θ via the maximum likelihood method seen in Section 7.1, that is basically deriving the log likelihood and seeing for what values it goes to zero.

$$\frac{\partial}{\partial\theta}\ln p(\mathcal{D}|\theta) = 0$$

$$\frac{\partial}{\partial\theta}\ln \theta^h(1-\theta)^t = 0$$

$$\frac{\partial}{\partial\theta}h\ln\theta + t\ln(1-\theta) = 0$$

$$h\frac{1}{\theta} + t\frac{1}{1-\theta}(-1) = 0$$

$$\frac{h}{\theta} = \frac{t}{1-\theta}$$

$$h - \theta h = t\theta$$

$$\theta = \frac{h}{h+t}$$

This tells us that h, t, that is the number of heads and tails in the dataset $\mathcal{D}$ respectively, are the sufficient parameters. This implies that for example we don't care about the order of the outcomes.
Let's now try to estimate the value of the parameter θ by using the Bayesian estimation. We have seen in Section 7.2 that the parameter posterior is proportional to:

$$P(\theta|\mathcal{D},\psi) \propto P(\mathcal{D}|\theta)P(\theta|\psi) \propto \theta^h(1\theta)^t\theta^{\alpha_h-1}(1-\theta)^{\alpha_t-1} = \theta^{h+\alpha_h-1}(1-\theta)^{t+\alpha_t-1}$$

That is the posterior is proportional to a Beta distribution with parameters $h+\alpha_h, t+\alpha_t$.
The prediction for a new event it's basically the expected value of the posterior Beta:

$$P(x|\mathcal{D}) = \int P(x|\theta)P(\theta|\mathcal{D},\psi)d\theta$$

The probability $P(x|\theta)$ is actually θ since the parameter represents the probability of success for an event.

$$P(x|\mathcal{D}) = \int \theta P(\theta|\mathcal{D},\psi)d\theta$$

Such integral is just the expected value of the Beta (Equation 5.8):

$$P(x|\mathcal{D}) = \mathrm{E}_{P(\theta|\mathcal{D},\psi)}[\theta] = \frac{h+\alpha_h}{h+t+\alpha_h+\alpha_t}$$

Our prior knowledge is encoded as a number $\alpha = \alpha_h + \alpha_t$ of imaginary experiments. α is called equivalent sample size. Notice that if $\alpha \to 0$, then the estimation is reduced to the classical maximum likelihood approach.

8.1.2 Multinomial distribution

Let's now consider an event with r states, that is r outcomes: $x \in \{x^1, \ldots, x^r\}$. For example tossing a six face dice is such an event with $r = 6$ states.
Such event can be modelled via a Multinomial distribution. To represent the outcomes we could use the one-hot encoding:

$$\boldsymbol{z}(x) = [z_1(x), \ldots, z_r(x)], z_k(x) = \begin{cases} 1 \text{ if } x = x^k \\ 0 \text{ otherwise} \end{cases}$$

For example if the outcome of rolling a dice was a 2, then $\boldsymbol{z}(x) = [0, 1, 0, 0, 0, 0]$.
The parameter is the vector $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_r]$ which contains the probability for each outcome.
Finally the probability mass function can be written as (Equation 5.4):

$$P(x|\boldsymbol{\theta}) = \prod_{k=1}^{r} \theta_k^{z_k(x)}$$

As from Table 8.1, the conjugate prior is a Dirichlet distribution (Equation 5.11):

$$P(\boldsymbol{\theta}|\psi) = P(\boldsymbol{\theta}|\alpha_1, \ldots, \alpha_r) = \frac{\Gamma(\alpha)}{\prod_{k=1}^{r} \Gamma(\alpha_k)} \prod_{k=1}^{r} \theta_k^{\alpha_k - 1}$$

Given a dataset $\mathcal{D}$ of N realizations, e.g., results of tossing a dice N times, then the likelihood function is:

$$P(\mathcal{D}|\boldsymbol{\theta}) = \prod_{j=1}^{N} \prod_{k=1}^{r} \theta_k^{z_k(x_j)} = \prod_{k=1}^{r} \theta_k^{N_k}$$

For example let's consider instead a dataset containing RGB values: $\mathcal{D} = \{R, R, R, G, B, G, B, R, B, G\}$.
The likelihood can be written as:

$$P(\mathcal{D}|\boldsymbol{\theta}) = \theta_R^4 \theta_G^3 \theta_B^3$$

Let's first apply maximum likelihood estimation:

$$\frac{\partial}{\partial \theta_k} \ln \prod_{k=1}^{r} \theta_k^{N_k} = 0$$

From which we obtain that:

$$\theta_k = \frac{N_K}{N}$$

Let's then apply Bayesian estimation. The parameter posterior is proportional to:

$$P(\boldsymbol{\theta}|\mathcal{D}, \psi) \propto P(\mathcal{D}|\boldsymbol{\theta}) P(\theta|\psi) \propto \prod_{k=1}^{r} \theta_k^{N_k + \alpha_k - 1}$$

From this is possible to observe that the posterior has a Dirichlet as expected from Table 8.1.
Finally let's compute the probability of a new event via Bayesian estimation:

$$P(x_k|\mathcal{D}) = \int \theta_k P(\boldsymbol{\theta}|\mathcal{D}, \psi) d\boldsymbol{\theta}$$

Which is exactly as the expected value for the Dirichlet distribution (Equation 5.12:

$$P(x_k|\mathcal{D}) = \mathrm{E}_{P(\boldsymbol{\theta}|\mathcal{D},\psi)}[\theta_k] = \frac{\alpha_k + N_k}{\sum_{i=0}^{r} (\alpha_i + N_i)} = \frac{N_k + \alpha_k}{N + \alpha}$$

9 Bayesian Networks

Bayesian Networks are a fast and easy way to create graphical models to represent more variables and their relations.
In general *probabilistic graphical models* are graphical representation of the *qualitative*[1] aspects of probability distributions allowing to:

- Visualize the structure of a probabilistic model in a simple and intuitive way, in particular the relations between the variables.
- Detect dependency or independency between variables without having to apply derivation rules.
- Express complex computations for inference and learning in terms of graphical manipulations.
- Represent multiple probability distributions with the same graph, abstracting from their quantitative aspects (e.g. discrete vs continuous distributions).

9.1 Structure

A Bayesian Network structure $\mathcal{G}$ is directed graph (graphical model).
Each node represents a random variable and each edge represents direct dependency between two random variables, that is one variable that is influenced by the other. This implies also that the father depends on the child, while the second does not depend on the first.
The structure hence can be described as encoding the independences assumptions:

$$\mathcal{I}_\ell(\mathcal{G}) = \{\forall i \; x_i \perp NonDecendants_{x_i} | Parents_{x_i}\}$$

NonDecendants are all those nodes that are not directly reachable from a node, that is the parents plus all the nodes that are not in its subtree. Let's consider node x_4 in Figure 9.1, its *NonDecendants* are its parents $\{x_1, x_2, x_3\}$ plus x_5 which is a non reachable node from x_4. In total the *NonDecendants* nodes are: $\{x_1, x_2, x_3, x_5\}$ The $\mathcal{I}_\ell$ dependency is said to be local ($_\ell$) since it's defined only wrt to a single variable.
A part from the structure, we need also to have a probability distribution. Let's consider a dataset $\mathcal{D}$ in which all variables are tied with the other by a distribution p which is a joint distribution. We'd like to represent qualitatively with a graph such distribution.
Since the Bayesian Network depends on the independences, it's possible to create a set $\mathcal{I}(p)$ of these by looking at the distribution p.
$\mathcal{G}$ is an *independency map* (**I-map**) for p if p satisfies the local independencies in $\mathcal{G}$:

$$\mathcal{I}_\ell(\mathcal{G}) \subseteq \mathcal{I}(p)$$

[1] The quantitative aspect is represented by the probabilistic distributions.

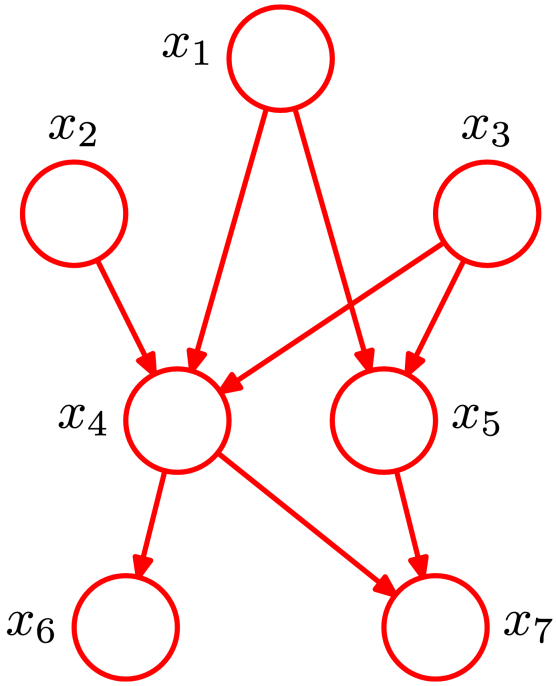


Figure 9.1: Example of Bayes Network.

The sufficient condition for $\mathcal{G}$ to be valid is that all the independences are also in p, while the opposite is not always true. Indeed, if some independence from p cannot be modelled into $\mathcal{G}$ via qualitative definition, then they must be modelled quantitatively.
Now we can describe p in terms of the graphical model, that is we can **factorize** the distribution based on the structure of the model.

Theorem 9.1

p is said to factorize according to $\mathcal{G}$ if:

$$p(x_1, \ldots, x_m) = \prod_{i=1}^{m} p(x_i | Parents_{x_i}) \tag{9.1}$$

Mind that this is a double implication: if $\mathcal{G}$ is an I-map for p, then p factorizes according to $\mathcal{G}$, then it's true also that if p factorizes according to $\mathcal{G}$, then $\mathcal{G}$ is an I-map for p. This can be proven as follows.

Proof. I-map⇒ factorization
If $\mathcal{G}$ is an I-map for p, then p satisfies at least these local independences:

$$\{\forall i \; x_i \perp NonDecendants_{x_i} | Parents_{x_i}\}$$

It's possible to order the variable in a topological order relative to $\mathcal{G}$, i.e.:

$$x_i \to x_j \Rightarrow i < j$$

That is the parents have a lower id than the children. Let us now decompose the joint probability using the chain rule as:

$$p(x_1, \ldots, x_m) = \prod_{i=1}^{m} p(x_i | x_1, \ldots, x_{i-1})$$

Finally local independences imply that for each x_i:

$$p(x_i|x_1,\dots,x_{i-1}) = p(x_i|Parents_{x_i})$$

Which by substitution:

$$p(x_1,\dots,x_m) = \prod_{i=1}^{m} p(x_i|Parents_{x_i})$$

□

Now we need to prove the inverse:

Proof. Factorization⇒ I-map
If p factorizes according to $\mathcal{G}$, the joint probability can be written as

$$p(x_1,\dots,x_m) = \prod_{i=1}^{m} p(x_i|Parents_{x_i})$$

Said $\mathcal{X}$ the set of all variables x_i: $\mathcal{X} = \{x_1,\dots,x_m\}$, let's consider variable x_m (repeat the following steps for the other variable), it's possible to write by the product and sum rules:

$$p(x_i|x_1,\dots,x_{m-1}) = \frac{p(x_1,\dots,x_m)}{p(x_1,\dots,x_{m-1})} = \frac{p(x_1,\dots,x_m)}{\sum_{x_m} p(x_1,\dots,x_m)}$$

Applying factorisation and isolating the only term containing x_m we get:

$$p(x_i|x_1,\dots,x_{m-1}) = \frac{\prod_{i=1}^{m} p(x_i|Parents_{x_i})}{\sum_{x_m}\prod_{i=1}^{m} p(x_i|Parents_{x_i})} =$$

$$= \frac{p(x_m|Parents_{x_m})\cancel{\prod_{i=1}^{m-1} p(x_i|Parents_{x_i})}}{\cancel{\prod_{i=1}^{m-1} p(x_i|Parents_{x_i})}\cancel{\sum_{x_m} p(x_m|Parents_{x_m})}}$$

The sum cancels out because summing all the value of a variable equals to 1.
Hence:

$$p(x_i|x_1,\dots,x_{m-1}) = p(x_m|Parents_{x_m})$$

□

For example consider Figure 9.1. It's possible to write:

$$p(x_1,\dots x_7) = p(x_1)p(x_2)p(x_3)p(x_4|x_1,x_2,x_3)p(x_5|x_2,x_3)p(x_6|x_4)p(x_7|x_4,x_5)$$

Mind that the joint probability $p(x_1,\dots,x_7)$ is largely influenced by the probability of node 4: $p(x_4|x_1,x_2,x_3)$. Now let's suppose that each variable defined a simple binary value, then the total number of possibility would be 2^7, against the 2^4 we found right now. This allows to be able to work with a lot of variables.

Definition 9.1: Bayesian Network

A Bayesian Network is a pair $(\mathcal{G}, p)$ where p factorizes over $\mathcal{G}$ and its represents as a set of conditional probability distribution associated with the nodes of $\mathcal{G}$

9.1.1 Example

Let's consider another example: genes A and B have independent prior probabilities and gene C can be enhanced by both A and B, with the following probability tables:

Gene	Value	$P(value)$
A	Active	0.3
A	Inactive	0.7

Gene	Value	$P(value)$
B	Active	0.3
B	Inactive	0.7

		A			
		Active		Inactive	
		B		B	
		Active	Inactive	Active	Inactive
C	Active	0.9	0.6	0.7	0.1
C	Inactive	0.1	0.4	0.3	0.9

Table 9.1: Table showing $p(A, B, |C)$.

9.1.2 Conditional Independence

First of let's recall Definition 5.7: two variables a, b are said to be independent written $a \perp b|\emptyset$ if:

$$p(a, b) = p(a)p(b)$$

Definition 9.2: Conditional Independence

Two variables a, b are conditionally independent given c, written $a \perp b|c$ if:

$$p(a, b|c) = p(a|c)p(b|c)$$

Graphical models allow to directly verify them through the **d-separation** criterion.

9.2 D-separation

Looking at some simpler graphs, some dependency rules can be inferred and then used on larger graphs by applying them to subgraphs.

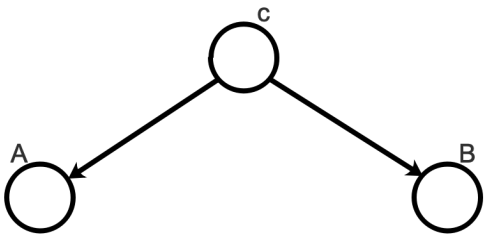
9.2.1 Two Nodes

Given two nodes, or they are independent, hence they are not connected, or they are dependent, hence they are connected.

9.2.2 Three Nodes

9.2.2.1 Tail-To-Tail

Also known as *common cause*.



As for the factorization Equation 9.1, the joint distribution can be expressed as:

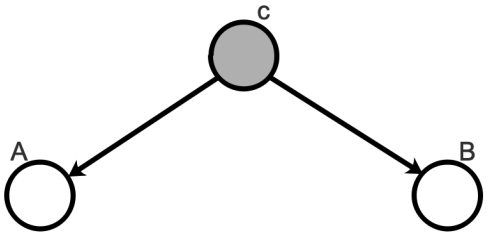
$$p(a, b, c) = p(a|c)p(b|c)p(c)$$

The tail-to-tail case says that a and b are *not independent*, written $a \not\perp\!\!\!\perp b$. If c is not given then we have that:

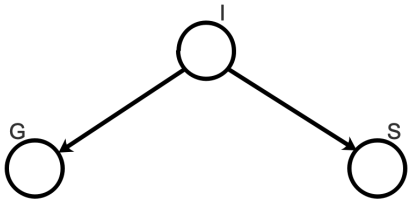
$$p(a, b) = \sum_c p(a|c)p(b|c)p(c) \neq p(a)p(b)$$

On the contrary, if c is given, then they are *conditionally independent*:

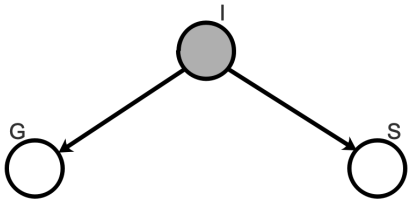
$$p(a, b|c) = \frac{p(a, b, c)}{p(c)} = p(a|c)p(b|c)$$



Let's consider an example: a company needs to choose the most intelligent (I) student between more students. For each student the variables grade (G) and score (S) are given. The following graph can be drawn since if the student is intelligent, then it should be seen in both grades and scores.



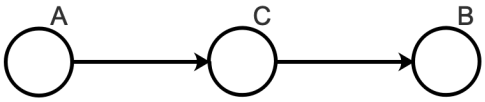
G and S are directly correlated, for example observing high scores could be indication of having a higher intelligence and hence also higher grades.



G and S are not directly dependent since once intelligence has been observed, the correlation disappear and nor G gives more information S, nor S gives more information about G than the what we already know.

9.2.2.2 Head-To-Tail

Also known as *indirect causal effect*.



The joint distribution can be expressed as:

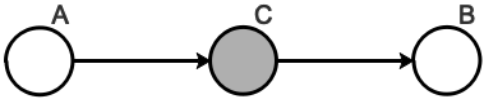
$$p(a, b, c) = p(b|c)p(c|a)p(a) = p(b|c)p(a|c)p(c)$$

If c is not given, then then a and b are *not independent*:

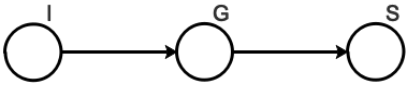
$$p(a, b) = p(a) \sum_c p(b|c)p(c|a) \neq p(a)p(b)$$

If instead c is given, then a and b are conditionally independent:

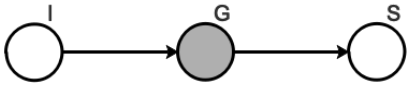
$$p(a, b|c) = \frac{p(b|c)p(a|c)p(c)}{p(c)} = p(b|c)p(a|c)$$



As before, let's consider the example of the intelligence of a student.



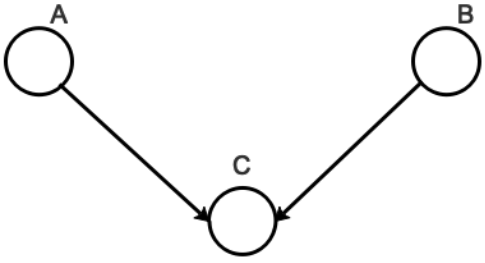
Intuitivly, if we observe that the student is intelligent, then we are more inclined to believe that it's grades G are good and that they will have a better score at the interview S, that is the probability of these latter events is higher conditioned on the observation that the student is intelligent.



Instead if we assume that G is observed, then it's intelligence no longer influences the score of the interview.

9.2.2.3 Head-To-Head

Also known as *common effect*.



The joint distribution can be expressed as:

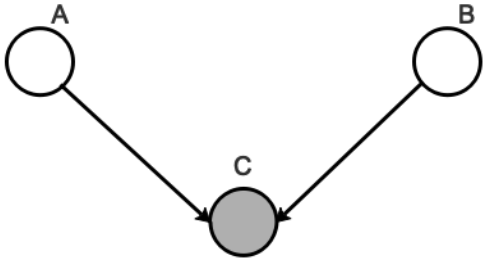
$$p(a, b, c) = p(c|a, b)p(a)p(b)$$

If c is not given, then then a and b are *independent*:

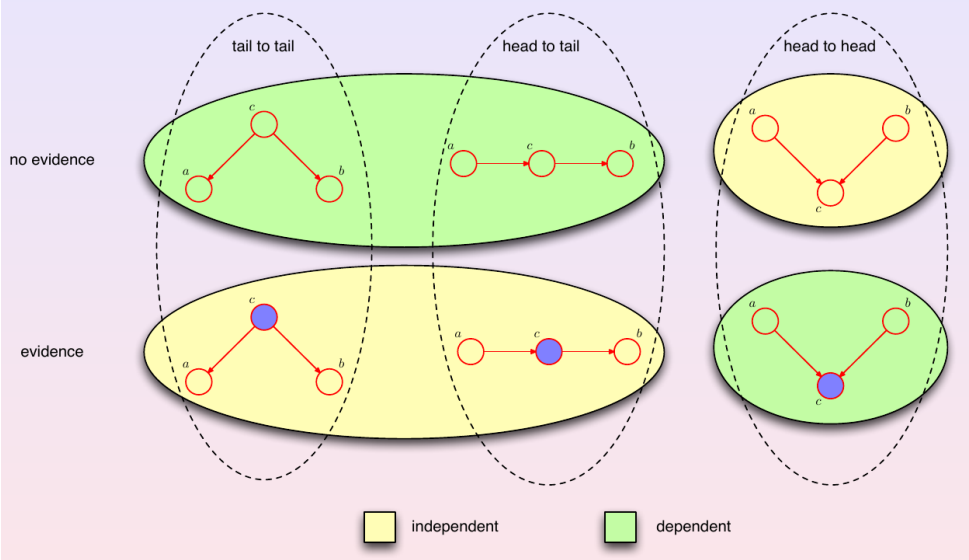
$$p(a,b) = \sum_c p(c|a,b)p(a)p(b) = p(a)p(b)$$

If instead c is given, then a and b are not conditionally independent, hence they are conditionally dependent:

$$p(a,b|c) = \frac{p(c|a,b)p(a)p(b)}{p(c)} \neq p(a|c)p(b|c)$$



Let's consider one last time the example of the student that is to be hired from a company. We have said that if c is not given, then a and b are independent, while if c is given they are dependent. Indeed, if c was the score of a test, then the student (G) and it was low, we could think that they are actually not that smart, but then if we were to observe that the test was difficult, then we could think that actually they are not *not* intelligent. Hence, a and b are actually correlated if c is given.



9.2.2.3.1 Example Head-To-Head Example

Let's consider a fuel system of a car. The fuel system is make of:

- Battery B: can either be charged ($B = 1$) or flat ($B = 0$);
- Fuel tank F: can either be full ($F = 1$) or empy ($F = 0$);
- Electric fuel gauge G: either full ($G = 1$) or empty ($G = 0$).

Let's now say that the probability of the battery to be charged is $P(B = 1) = 0.9$ and the probability for the fuel tank to be full is $P(F = 1) = 0.9$.
Since he electric fuel gauge is conditioned on both:

we can derive:

$$P(G=1|B=1,F=1)=0.8 \quad P(G=1|B=1,F=0)=0.2$$
$$P(G=1|B=0,F=1)=0.2 \quad P(G=1|B=0,F=0)=0.1$$

10 17/10/2019

$P(F = 0|G = 0)$ non è direttamente presente nelle tabelle e quindi bisogna riuscire a ricrearla. Per esempio abbiamo una probabilità: $P(G = 0|F = 0)$ e quindi possiamo usare Bayes per girarla:

$$P(F = 0|G = 0) = \frac{P(G = 0|F = 0)P(F = 0)}{P(G = 0)}$$

Di questo conosimamo solo P(F=0), tutto il resto ce lo calcoliamo:

$$P(G = 0|F = 0) = \sum_{B\in\{0,1\}} P(G = 0, |F = 0) = \sum_{B\in\{0,1\}} P(G = 0|B, F = 0)P(B|F = 0)$$

La prima probabilità la abbiamo, la seconda non l'abbiamo, però nella struttura dell'esempio B e F sono indipendenti e quindi $P(B|F = 0) = P(B)$. quindi otteniamo:

$$= P(G = 0|B = 0, F = 0)P(B = 0) + P(G = 0|B = 1, F = 0)P(B = 1)$$

Ora queste probabilità le conosciamo quindi:

$$= 0.9 + 0.1 + (1 - 0.2) + 0.9$$

Attenzione che se osserviamo solo B e F i due sono indipendenti, se invece consideriamo tutta la rete, allora abbiamo G e B e F sono dipendenti. Mind that also saying given doesn't mean we know the value for sure, it just means we assume it gets that value.
Dobbiamo ora calcolarci $P(G = 0)$.

$$P(G = 0) = \sum_{F,B} P(G = 0, F, B) = \sum_{F,B} P(G = 0|F, B)P(F)P(B)$$

Di queste probabilità abbiamo tutti i dati quindi possiamo concludere.

10.1 d-separation

Abbiamo visto le 3 diverse strutte nel caso di 3 nodi. Consider tail to tail and for example earthquake->alarm, burglar->alarm if you see the alarm you can think of the burglar, but if you see earthquake, than for sure it's not a burglar, so the two fathers are related.
If we add a fourth node phone call from alarm, then

10.1.1 Definition

Consideriamo un grafo come quello in figura. Vogliamo fare dei calcoli arbitrari sulla dipendenza, per esempio che A=x1, x2 sono indiependenti da B=x5 e x7 e un altro set $C = \emptyset$. Possiamo usare le regole che abbiamo visto adesso. Praticamente per ogni nodo in un set dobbiamo vedere tutti i possibbili percorsi che ci portano a un altro set e sfruttando le regole di prima possiamo dire

che non esiste un passo se ad esempio abbiamo tail to tail o head to tail con evidence nel mezzo, oppure head to head senza evidence. So x4->x6 is head to tail, non ho informazione quindi è libero, x4,x5,x6 è head to head e non c'è evidence quindi è bloccato. Consideriamo il percorso x1x4x6x7, i primi 3 sono head to tail senza evidence quindi posso passare, e anche dopo x4x6x7. Supponiao ora che C contenga x6, allora il path 3 è bloccato, mentre il path 1 non è più bloccato perchè x6 diventa evidence.
C quindi è il set con l'evidence.

10.2 BN independences

Abbiamo visto le indipendenze locali, ossia indiepndenze dei nodi rispetto a tutto il resto. A questo si aggiungono indipendenze globali (che racchiudono anche le primme), definire come in slide.

10.3 Equivalence classes

Possiamo raggruppare le indipendenze in classi, ad esempio head-to-head in una direzione o nell'altra producono la stessa cosa, come anche tail to tail nel caso senza evidence. Due strutture sono nella stessa classe si equivalenza se codificano la stessa dipendenza. In principio ogni struttura da una classe è equivalente per rappresentare i dati.

10.4 I-maps

Per riuscire ad avere un buon modello bisogna riuscire a codificareil amggior numero di strutture, quindi abbbiamo una definizone di mappa minima come quella in slide. Questo non soddisfa il fatto che riusciamo a prendere tutte le dipendenze.
Al contrario una mappa perfetta è una mappa che riesce a prendere tutte le dipendenze. Ovviamente una mappa perfetta è anche minima, ma non viceversa. Esiste un algoritmo per trovare una mappa perfetta per una distribuzione ed è esponenziale rispetto al numero di connessioni massime che un nodo ha. Ci sono della distribuzione che non hanno delle mappe perfette, per risolvere queste distribuzioni si usano dei grafi non diretti chiamati Markov Networks.

10.5 Making bayesian networks

Un esperto deve fornirci le variabili di interesse, ad esempio i sintomi e quello che potremmo voler trovare. A questo punto vogliamo costruire dei collegamenti e per fare questo vogliamo collegare varaibili che sono in relazione causale, questo ci permette di avere anche un grafo con meno collegamenti. A questo punto dobbiamo mettere le probabilità per ogni configurazione.
Quello che si fa di solito è prendere un sacco di dati e imparare i parametri e in alcuni casi anche la struttura (partendo comunque da una struttura minima).

10.6 Inference in graphical model

Un BN, come tutti i cosi grafici, modella una probabilità congiunta. Per questo motivo si ha:

$$P(X|E=e) = \frac{p(X, E=e)}{P(E=e)}$$

Vedi fogli (6) ma è troppo grande, quindi possiamo usare la scomposizione. La struttura più semplice che possiamo pensare è una catena dove ogni varaibile ha una connessione a quella dopo. In questo modo la probabilità di tutte le variabili nella reteè:

$$P(X) = P(x1)P(X2|x1)...P(x_N|x_{N-1})$$

Supponiamo di voler calcolare la probailità di una variabile senza condizione:

$$P(x_n) = \sum_{x_1} ... \sum_{x_N} P(X)$$

In questo modo ad esempio $\sum_{x_N}$ è solo la parte finale della formula di prima e tutto il resto invece è costante:

$$\mu_\beta(x_{n-1}) = \sum_{x_N} P(x_N|x_{N-1})$$

Se ora si sommano tutti i valori si ottiene 1, ma non sempre??????
Possiamo rsicrivere come:

$$\sum_{x_1} ... \sum_{x_{N-1}} P(x_1)P(x_2|x_1)...P(x_{n-1}|x_{n-2})\mu_\beta(x_{n-1})$$

A questo punto consideriamo la sommatoria per $n-1$ ed è la stessa cosa, tutto costante tranne l'ultima cosa:

$$\mu_\beta(x_{n-2}) = \sum$$

E continuo fino ad arrivare da qualche parte e si ha:

$$\mu_\beta(x_n) = \sum_{x_{n+1}} P(x_{n+1}|x_n)\mu_\beta(x_{n+1})$$

Ora manca qualche altra sommatora, cioè non ci si capisce più niente e andadno al contrario si ha:

$$\mu_\alpha(x_2) = \sum_{x_1} P(x_1)P(x_2|x_1)$$
$$\mu_\alpha(x_3) = \sum_{x_2} P(x_3|x_2)\mu_\alpha(x_2)$$

E alla fine si ha mettendo tutto insieme che:

$$p(X_n) = \mu_\alpha(X_n)\mu_\beta(X_n)$$

Migliorando le prestazioni per qualche altro motivo portandole da esponenziali a qualcos'altro.
Possiamo pensare a μ come un messaggio e mandarlo avanti α o indietro (backwards) β.
Supponiamo di avere fatto tutti i calcoli e mandati i messaggi e abbbiamo la nostra rete e dobbiamo computare X_{n+1} quindi dovremmo ricominciare a mandare tutti i messaggi, ma la maggior parte sono li stessi quindi non vogliamo ricalcolarli. Al posto di fare questo si manda un messaggio dall'inizio alla fine e dalla fine all'inizio e boh.
Questo è tutto senza evidence, ma di solito abbiamo anche questa. Dobbiamo quindi modiifcare l'algoritmo per tenere conto dell'evidence: abbiamo calcolato $P(x_n)$.
Questo funziona con le catene, ma non sono l'unico tipo di struttura. altri tipo di struttura sono gli alberi ossia strutture dove ogni nodo ha un solo genitore, uno che non ha padri che è la radice e nodi che non hanno figli che sono le foglie, oppure poly-tree: si ha comunque un solo percorso tra coppie di nodi. Anche gli undirected trees, questo perchè l'inferenza che abbiamo visto prima è valida sia per BN che per Markov Networks.

10.7 Factor graph

Il modo per parlare di queste strutture da catenre è passare per una struttura intermedia chiamata factor graph utile solo per fare gli inference graph.
Questo grafo ha un nodo per ogni variabbile e anche nodi (detti factor nodes) per ogni evidence (forse= che ha. È anche un grafo non direzionale.
Ogni nodo factor deve essere collegato a tutti i nodi delle probabilità da cui deriva la scomposizione.