# *Text Classification: the Geometrical approach. Vector models, and similarity*

R. Basili

Corso di *Web Mining* e *Retrieval*
a.a. 2021-22

March 14, 2022

# Outline

## Outline

# *Real-valued Vector Space*

### *Vector Space definition:*

A *vector space* is a set $V$ of objects called *vectors* $\underline{x} = \begin{pmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{pmatrix} = |\underline{x}\rangle$

where we can simply refer to a vector by $\underline{x}$, or using the specific realization called *column vector*, (*Dirac* notation $|\underline{x}\rangle$)

# *Real-valued Vector Space*

*Vector Space definition:*

A vector space need to satisfy the following axioms:

# *Real-valued Vector Space*

## *Vector Space definition:*

A vector space need to satisfy the following axioms:

### *Sum*

To every pair, $\underline{x}$ and $\underline{y}$, of vectors in $V$ there corresponds a vector $\underline{x}+\underline{y}$, called the sum of $\underline{x}$ and $\underline{y}$, in such a way that:

1. sum is commutative, $\underline{x}+\underline{y}=\underline{y}+\underline{x}$
2. sum is associative, $\underline{x}+\left(\underline{y}+\underline{z}\right)=\left(\underline{x}+\underline{y}\right)+\underline{z}$
3. there exist in $V$ a unique vector $\Phi$ (called the origin) such that $\underline{x}+\Phi=\underline{x}\ \forall\underline{x}\in V$
4. $\forall\underline{x}\in V$ there corresponds a unique vector $-\underline{x}$ such that $\underline{x}+\left(-\underline{x}\right)=\Phi$

# Real-valued Vector Space

## Vector Space definition:

A vector space need to satisfy the following axioms:

### Sum

To every pair, $\underline{x}$ and $\underline{y}$, of vectors in $V$ there corresponds a vector $\underline{x}+\underline{y}$, called the sum of $\underline{x}$ and $\underline{y}$, in such a way that:

1. sum is commutative, $\underline{x}+\underline{y}=\underline{y}+\underline{x}$
2. sum is associative, $\underline{x}+(\underline{y}+\underline{z})=(\underline{x}+\underline{y})+\underline{z}$
3. there exist in $V$ a unique vector $\Phi$ (called the origin) such that $\underline{x}+\Phi=\underline{x}\ \forall \underline{x}\in V$
4. $\forall \underline{x}\in V$ there corresponds a unique vector $-\underline{x}$ such that $\underline{x}+(-\underline{x})=\Phi$

### Scalar Multiplication

To every pair $\alpha$ and $\underline{x}$, where $\alpha$ is a scalar and $\underline{x}\in V$, there corresponds a vector $\alpha\underline{x}$, called the product of $\alpha$ and $\underline{x}$, in such a way that:

1. associativity $\alpha(\beta\underline{x})=(\alpha\beta)\underline{x}$
2. $1\underline{x}=\underline{x} \qquad \forall \underline{x}\in V$
3. mult. by *scalar* is distributive wrt. vector addition $\alpha(\underline{x}+\underline{y})=\alpha\underline{x}+\alpha\underline{y}$
4. mult. by *vector* is distributive wrt. scalar addition $(\alpha+\beta)\underline{x}=\alpha\underline{x}+\beta\underline{x}$

# Vector Operations

**Sum of two vector $\underline{x}$ and $\underline{y}$**

$$\underline{x}+\underline{y} = |\underline{x}\rangle + |\underline{y}\rangle = \begin{pmatrix} x_1 + y_1 \\ \cdot \\ \cdot \\ \cdot \\ x_n + y_n \end{pmatrix}$$

# Vector Operations

### Sum of two vector $\underline{x}$ and $\underline{y}$

$$\underline{x}+\underline{y}=|\underline{x}\rangle+|\underline{y}\rangle=\begin{pmatrix} x_1+y_1 \\ \cdot \\ \cdot \\ \cdot \\ x_n+y_n \end{pmatrix}$$

### Multiplication by scalar $\alpha$

$$\alpha\underline{x}=\alpha|\underline{x}\rangle=\begin{pmatrix} \alpha x_1 \\ \cdot \\ \cdot \\ \cdot \\ \alpha x_n \end{pmatrix}$$

### Linear combination

$$\underline{y}=c_1\underline{x}_1+\cdots+c_n\underline{x}_n$$

or

$$|\underline{y}\rangle=c_1|\underline{x}_1\rangle+\cdots+c_n|\underline{x}_n\rangle$$

# *Linear dependence*

### *Conditions for linear dependence*

A set o vectors $\{\underline{x}_1, \ldots, \underline{x}_n\}$ are *linearly dependent* if there a set constant scalars $c_1, \ldots, c_n$ exists, not all 0, such that:

$$c_1 \underline{x}_1 + \cdots + c_n \underline{x}_n = \underline{0}$$

# *Linear dependence*

## *Conditions for linear dependence*

A set o vectors $\{\underline{x}_1, \ldots, \underline{x}_n\}$ are *linearly dependent* if there a set constant scalars $c_1, \ldots, c_n$ exists, not all 0, such that:

$$c_1\underline{x}_1 + \cdots + c_n\underline{x}_n = \underline{0}$$

## *Conditions for linear independence*

A set o vectors $\{\underline{x}_1, \ldots, \underline{x}_n\}$ are *linearly independent* if and only if the *linear condition* $c_1\underline{x}_1 + \cdots + c_n\underline{x}_n = \underline{0}$ is satisfied only when $c_1 = c_2 = \cdots = c_n = 0$

# *Basis*

***Definition:***

A *basis* for a space is a set of $n$ linearly independent vectors in a $n$-dimensional vector space $V_n$.

# Basis

*Definition:*

A *basis* for a space is a set of $n$ linearly independent vectors in a $n$-dimensional vector space $V_n$.

This means that every arbitrary vector $\underline{x} \in V$ can be expressed as linear combination of the *basis* vectors,

$$\underline{x} = c_1\underline{x}_1 + \cdots + c_n\underline{x}_n$$

where the $c_i$ are called the co-ordinates of $\underline{x}$ wrt. the basis set $\{\underline{x}_1, \ldots, \underline{x}_n\}$

# Inner Product

*Definition:*

Is a real-valued function on the cross product $V_n \times V_n$ associating with each pair of vectors $(\underline{x}, \underline{y})$ a unique real number.
The function $(.,.)$ has the following properties:

1. $(\underline{x}, \underline{y}) = (\underline{y}, \underline{x})$
2. $(\underline{x}, \lambda \underline{y}) = \lambda (\underline{x}, \underline{y})$
3. $(\underline{x}_1 + \underline{x}_2, \underline{y}) = (\underline{x}_1, \underline{y}) + (\underline{x}_2, \underline{y})$
4. $(\underline{x}, \underline{x}) \geq 0$ and $(\underline{x}, \underline{x}) = 0$ **iff** $\underline{x} = \underline{0}$

# Inner Product

**Definition:**

Is a real-valued function on the cross product $V_n \times V_n$ associating with each pair of vectors $(\underline{x}, \underline{y})$ a unique real number.
The function $(.,.)$ has the following properties:

1. $(\underline{x}, \underline{y}) = (\underline{y}, \underline{x})$
2. $(\underline{x}, \lambda \underline{y}) = \lambda (\underline{x}, \underline{y})$
3. $(\underline{x}_1 + \underline{x}_2, \underline{y}) = (\underline{x}_1, \underline{y}) + (\underline{x}_2, \underline{y})$
4. $(\underline{x}, \underline{x}) \geq 0$ and $(\underline{x}, \underline{x}) = 0$ **iff** $\underline{x} = \underline{0}$

**Standard Inner Product**

$$(\underline{x}, \underline{y}) = \sum_{i=1}^{n} x_i y_i$$

# Inner Product

## Definition:

Is a real-valued function on the cross product $V_n \times V_n$ associating with each pair of vectors $(\underline{x}, \underline{y})$ a unique real number.
The function $(.,.)$ has the following properties:

1. $(\underline{x}, \underline{y}) = (\underline{y}, \underline{x})$
2. $(\underline{x}, \lambda \underline{y}) = \lambda (\underline{x}, \underline{y})$
3. $(\underline{x}_1 + \underline{x}_2, \underline{y}) = (\underline{x}_1, \underline{y}) + (\underline{x}_2, \underline{y})$
4. $(\underline{x}, \underline{x}) \geq 0$ and $(\underline{x}, \underline{x}) = 0$ **iff** $\underline{x} = \underline{0}$

## Standard Inner Product

$$(\underline{x}, \underline{y}) = \sum_{i=1}^{n} x_i y_i$$

## Other notations

- $\underline{x}^T \underline{y}$ where $\underline{x}^T$ is the transpose of $\underline{x}$
- $\langle \underline{x} | \underline{y} \rangle$ or sometimes $\langle \underline{x} || \underline{y} \rangle$ in Dirac notation

# *Norm*

### *Geometric interpretation*

Geometrically the *norm* represent the length of the vector

# Norm

**Geometric interpretation**

Geometrically the *norm* represent the length of the vector

**Definition**

The *norm* id a function $||.||$ from $V_n$ to $\mathbb{R}$

# Norm

**Geometric interpretation**

Geometrically the *norm* represent the length of the vector

**Definition**

The *norm* id a function $||.||$ from $V_n$ to $\mathbb{R}$

**Euclidean Norm:**

$$||\underline{x}|| = \sqrt{(\underline{x},\underline{x})} = \sqrt{\sum_{i=1}^{n} x_i^2} = \left(x_1^2 + \cdots + x_n^2\right)^{1/2}$$

# Norm

### Geometric interpretation

Geometrically the *norm* represent the length of the vector

### Definition

The *norm* id a function $||.||$ from $V_n$ to $\mathbb{R}$

### Euclidean Norm:

$$||\underline{x}|| = \sqrt{(\underline{x},\underline{x})} = \sqrt{\sum_{i=1}^{n} x_i^2} = \left(x_1^2 + \cdots + x_n^2\right)^{1/2}$$

### Properties

1. $||\underline{x}|| \geq 0$ and $||\underline{x}|| = 0$ if and only if $\underline{x} = 0$
2. $||\alpha \underline{x}|| = |\alpha| \, ||\underline{x}||$ for all $\alpha$ and $\underline{x}$
3. $\forall \underline{x}, \underline{y}, |(\underline{x}, \underline{y})| \leq ||\underline{x}|| \, ||\underline{y}||$ (Cauchy-Schwartz)

# Norm

**Geometric interpretation**

Geometrically the *norm* represent the length of the vector

**Definition**

The *norm* id a function $||.||$ from $V_n$ to $\mathbb{R}$

**Euclidean Norm:**

$$||\underline{x}|| = \sqrt{(\underline{x},\underline{x})} = \sqrt{\sum_{i=1}^{n} x_i^2} = \left(x_1^2 + \cdots + x_n^2\right)^{1/2}$$

**Properties**

1. $||\underline{x}|| \geq 0$ and $||\underline{x}|| = 0$ if and only if $\underline{x} = 0$
2. $||\alpha\underline{x}|| = |\alpha|\,||\underline{x}||$ for all $\alpha$ and $\underline{x}$
3. $\forall \underline{x}, \underline{y}, |(\underline{x},\underline{y})| \leq ||\underline{x}||\,||\underline{y}||$ (Cauchy-Schwartz)

# Norm

### Geometric interpretation

Geometrically the *norm* represent the length of the vector

### Definition

The *norm* id a function $||.||$ from $V_n$ to $\mathbb{R}$

### Euclidean Norm:

$$||\underline{x}|| = \sqrt{(\underline{x},\underline{x})} = \sqrt{\sum_{i=1}^{n} x_i^2} = \left(x_1^2 + \cdots + x_n^2\right)^{1/2}$$

### Properties

1. $||\underline{x}|| \geq 0$ and $||\underline{x}|| = 0$ if and only if $\underline{x} = 0$
2. $||\alpha \underline{x}|| = |\alpha| \, ||\underline{x}||$ for all $\alpha$ and $\underline{x}$
3. $\forall \underline{x}, \underline{y}, |(\underline{x},\underline{y})| \leq ||\underline{x}|| \, ||\underline{y}||$ (Cauchy-Schwartz)

A vector $\underline{x} \in V_n$ is a *unit vector*, or *normalsized*, when $||\underline{x}|| = 1$

# *From Norm to distance*

In $V_n$ we can define the distance between two vectors $\underline{x}$ and $\underline{y}$ as:

$$d(\underline{x},\underline{y}) = ||\underline{x}-\underline{y}|| = \sqrt{(\underline{x}-\underline{y},\underline{x}-\underline{y})} = \left((x_1-y_1)^2+\cdots+(x_n-y_n)^2\right)^{1/2}$$

These measure, noted sometimes as $||\underline{x}-\underline{y}||_2^2$, is also named *Euclidean distance*.

# From Norm to distance

In $V_n$ we can define the distance between two vectors $\underline{x}$ and $\underline{y}$ as:

$$d(\underline{x},\underline{y}) = ||\underline{x}-\underline{y}|| = \sqrt{(\underline{x}-\underline{y},\underline{x}-\underline{y})} = \left((x_1-y_1)^2+\cdots+(x_n-y_n)^2\right)^{1/2}$$

These measure, noted sometimes as $||\underline{x}-\underline{y}||_2^2$, is also named *Euclidean distance*.

*Properties:*

- $d(\underline{x},\underline{y}) \geq 0$ and $d(\underline{x},\underline{y}) = 0$ if and only if $\underline{x}=\underline{y}$
- $d(\underline{x},\underline{y}) = d(\underline{y},\underline{x})$ symmetry
- $d(\underline{x},\underline{y}) =\leq d(\underline{x},\underline{z}) + d(\underline{z},\underline{y})$ triangle inequality

# From Norm to distance

An immediate consequence of Cauchy-Schwartz property is that:

$$-1 \leq \frac{(\underline{x},\underline{y})}{||\underline{x}||\,||\underline{y}||} \leq 1$$

and therefore we can express it as:

$$(\underline{x},\underline{y}) = ||\underline{x}||\,||\underline{y}|| \cos\varphi \qquad 0 \leq \varphi \leq \pi$$

where $\varphi$ is the angle between the two vectors $\underline{x}$ and $\underline{y}$

# From Norm to distance

An immediate consequence of Cauchy-Schwartz property is that:

$$-1 \leq \frac{(\underline{x},\underline{y})}{||\underline{x}||\,||\underline{y}||} \leq 1$$

and therefore we can express it as:

$$(\underline{x},\underline{y}) = ||\underline{x}||\,||\underline{y}||\cos\varphi \qquad 0 \leq \varphi \leq \pi$$

where $\varphi$ is the angle between the two vectors $\underline{x}$ and $\underline{y}$

## Cosine distance

$$\cos\varphi = \frac{(\underline{x},\underline{y})}{||\underline{x}||\,||\underline{y}||} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} y_i^2}}$$

# From Norm to distance

An immediate consequence of Cauchy-Schwartz property is that:

$$-1 \leq \frac{(\underline{x},\underline{y})}{||\underline{x}||\,||\underline{y}||} \leq 1$$

and therefore we can express it as:

$$(\underline{x},\underline{y}) = ||\underline{x}||\,||\underline{y}||\cos\varphi \qquad 0 \leq \varphi \leq \pi$$

where $\varphi$ is the angle between the two vectors $\underline{x}$ and $\underline{y}$

## Cosine distance

$$\cos\varphi = \frac{(\underline{x},\underline{y})}{||\underline{x}||\,||\underline{y}||} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} y_i^2}}$$

If the vectors $\underline{x}$, $\underline{y}$ have the norm equal to 1 then:

$$\cos\varphi = \sum_{i=1}^{n} x_i y_i = (\underline{x},\underline{y})$$

# Orthogonality

## Definition

$\underline{x}$ and $\underline{y}$ are orthogonal if and only if $(\underline{x}, \underline{y}) = 0$

## Orthonormal basis

A set of linearly independent vectors $\{\underline{x}_1, \ldots, \underline{x}_n\}$ constitutes an orthonormal basis for the space $V_n$ if and only if

$$(\underline{x}_i, \underline{x}_j) = \delta_{ij} = \left( \begin{array}{ccc} 1 & \text{if} & i = j \\ 0 & \text{if} & i \neq j \end{array} \right)$$

# *Similarity*

*Looking to texts as points a n-dimensional space*

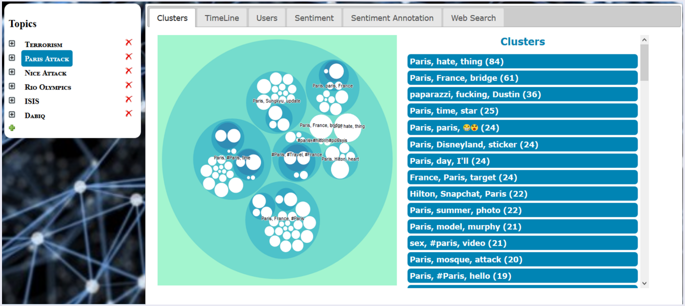
A structure for organizing large bodies of texts for efficient searching and browsing can be the notion of metric space.
Internet search engines may suitably exploit cluster analysis to documents in order to organize them visually.

# Similarity

## *Looking to texts as points a n-dimensional space*

A structure for organizing large bodies of texts for efficient searching and browsing can be the notion of metric space.
Internet search engines may suitably exploit cluster analysis to documents in order to organize them visually.

## *Clustering of texts for browsing*



# *Similarity*

### *Looking to texts as points a n-dimensional space*

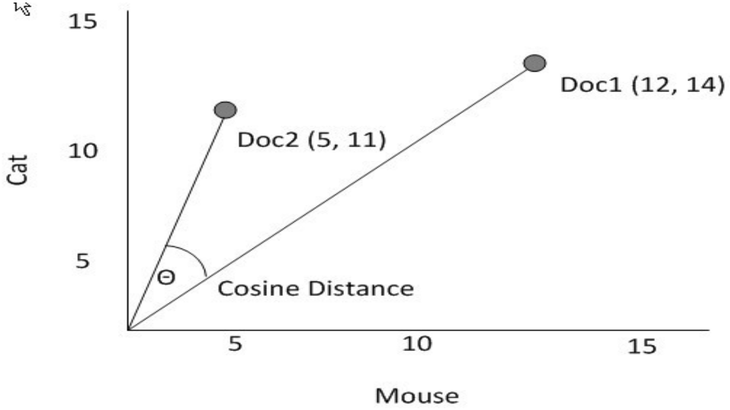
A structure for organizing large bodies of texts for efficient searching and browsing can be the notion of metric space.
Internet search engines may suitably exploit cluster analysis to documents in order to organize them visually.

### *Document and vectors*

A document is commonly represented as a *vector* consisting of the suitably normalized frequency counts of words or terms.
Each document typically contains only a small percentage of all the words ever used. If we consider each document as a multi-dimensional vector and then try to cluster documents based on their word contents, the problem differs from classic clustering scenarios in several ways.

# *Text as Vectors*

In Vector Space Model documents words corresponds to the space (orthonormal) basis, and individual texts are mapped into vectors ...



# Text Classification in the Vector Space Model

## Text Classification: Definition

Given:

- a set of target categories, $C = \{C_1, ..., C_n\}$:
- the set T of documents,

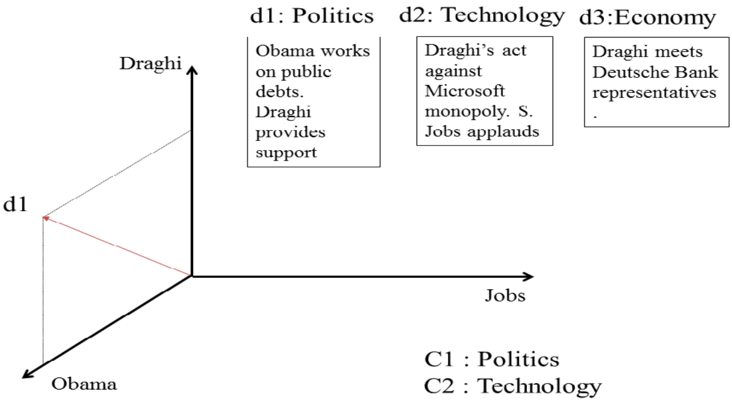
define a function: $f : T \leftarrow 2^C$

## Vector Space Model (Salton89)

Features are dimensions of a Vector Space.
Documents $d$ and Categories $C_i$ are mapped to vectors of feature weights ($\underline{d}$ and $\underline{C}_i$, respectively).
**Geometric Model of $f()$:**
A document $d$ is assigned to a class $C_i$ if $(\underline{d}, \underline{C}_i) > \tau_i$

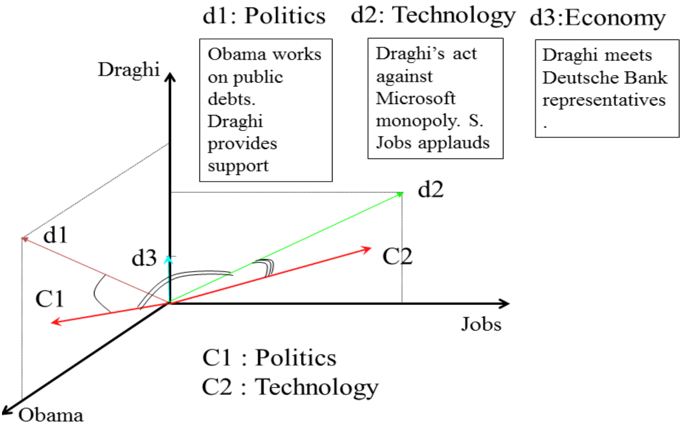
# Text Classification: Vector Space Modeling

In Vector Space Model documents words corresponds to the space (orthonormal) basis, and individual texts are mapped into vectors ...



# Text Classification: Classification Inference

Categories are also vectors and consine similarity measures can support the final inference about category membership, e.g. $d1 \in C1$ and $d2 \in C2$:



# *A simple model for Text Classification*

## *Motivation*

**Rocchio's** is one of the first and simple models for *supervised text classification* where:

- *document vectors* are weighted according to a standard function, called $tf \cdot idf$,

# A simple model for Text Classification

## *Motivation*

**Rocchio's** is one of the first and simple models for *supervised text classification* where:

- *document vectors* are weighted according to a standard function, called $tf \cdot idf$,
- *category vectors*, $\underline{C}_1, \ldots, \underline{C}_n$, are obtained by *averaging* the behaviour of the training examples.

# A simple model for Text Classification

## *Motivation*

**Rocchio's** is one of the first and simple models for *supervised text classification* where:

- *document vectors* are weighted according to a standard function, called $tf \cdot idf$,
- *category vectors*, $\underline{C}_1, \ldots, \underline{C}_n$, are obtained by *averaging* the behaviour of the training examples.

We thus need to define a weighting function: $\omega(w, d)$ for individual words $w$ in documents $d$ and a method to design a category vector, i.e. a profile, as a linear combination of document vectors.

# A simple model for Text Classification

## Motivation

**Rocchio's** is one of the first and simple models for *supervised text classification* where:

- *document vectors* are weighted according to a standard function, called $tf \cdot idf$,
- *category vectors*, $\underline{C}_1, \ldots, \underline{C}_n$, are obtained by *averaging* the behaviour of the training examples.

We thus need to define a weighting function: $\omega(w,d)$ for individual words $w$ in documents $d$ and a method to design a category vector, i.e. a profile, as a linear combination of document vectors.

## Similarity

Once vectors for documents and Category profiles $(\underline{C}_i)$ are made available than the standard cosine similarity is adopted for inferencing, i.e. again a document $d$ is assigned to a class $C_i$ if $(\underline{d}, \underline{C}_i) > \tau_i$

# *Term weighting through tf · idf*

Every term $w$ in a document $d$, as a feature $f$, receives a weight in the vector representation $\underline{d}$ that accounts for the occurrences of $w$ in $d$ as well as the occurrences in other documents of the collection.

## *Definition*

A word $w$ has a weight $\omega(w,d)$ in a document $d$ defined as

$$\omega(w,d) = \omega_w^d = o_w^d \cdot log\frac{N}{N_w}$$

where:

- $N$ is the overall number of documents,
- $N_w$ is the number of documents that contain the word $w$ and

# Term weighting through $tf \cdot idf$

Every term $w$ in a document $d$, as a feature $f$, receives a weight in the vector representation $\underline{d}$ that accounts for the occurrences of $w$ in $d$ as well as the occurrences in other documents of the collection.

## *Definition*

A word $w$ has a weight $\omega(w,d)$ in a document $d$ defined as

$$\omega(w,d) = \omega_w^d = o_w^d \cdot log \frac{N}{N_w}$$

where:

- $N$ is the overall number of documents,
- $N_w$ is the number of documents that contain the word $w$ and
- $o_w^d$ is the number of occurrences of $w$ in $d$

# Term weighting through $tf \cdot idf$

Every term $w$ in a document $d$, as a feature $f$, receives a weight in the vector representation $\underline{d}$ that accounts for the occurrences of $w$ in $d$ as well as the occurrences in other documents of the collection.

## Definition

A word $w$ has a weight $\omega(w,d)$ in a document $d$ defined as

$$\omega(w,d) = \omega_w^d = o_w^d \cdot log\frac{N}{N_w}$$

where:

- $N$ is the overall number of documents,
- $N_w$ is the number of documents that contain the word $w$ and
- $o_w^d$ is the number of occurrences of $w$ in $d$

# *Term weighting through $tf \cdot idf$*

The weight $\omega_w^d$ of term $w$ in document $d$ is called $tf \cdot idf$ as:

## *Term Frequency, $tf_w^d$*

The term frequency $o_w^d$ emphasize terms that are cally relevant for a document. Its normalizd version

$$tf_w^d = \frac{o_w^d}{max_{x \in d} o_x^d}$$

is often employed.

# *Term weighting through tf · idf*

The weight $\omega_w^d$ of term $w$ in document $d$ is called $tf \cdot idf$ as:

### *Term Frequency, $tf_w^d$*

The term frequency $o_w^d$ emphasize terms that are cally relevant for a document. Its normalizd version

$$tf_w^d = \frac{o_w^d}{max_{x \in d} o_x^d}$$

is often employed.

### *Inverse Document Frequency, $idf_w$*

The inverse document frequency $log \frac{N}{N_w}$ emphasizes only terms that are relatively not frequent in the corpus, by discarding common words that are not characterizing any specific subset of a collection. Notice how when $w$ occurs in *every* document $d$ then $N_w = N$ so that $idf_w = log \frac{N}{N_w} = 0$

# Representing Categories: the Rocchio model

The last step in providing a geometric account of text categorization is related to the represetation of a category $C_i$.

## Definition: Category Profile

A word $w$ has a weight $\Omega(w, C_i)$ in a document category vector $\underline{C}_i$ defined as:

$$\Omega(w, C_i) = \Omega_w^i = max\left\{0, \frac{\beta}{|T_i|}\sum_{d \in T_i} \omega_w^d - \frac{\gamma}{|\overline{T_i}|}\sum_{d \in \overline{T_i}} \omega_w^d\right\}$$

where $T_i$ is the set of training documents classified in $C_i$ and $\overline{T_i}$ are the set of training document not classified in $C_i$

# *Representing Categories: the Rocchio model*

The last step in providing a geometric account of text categorization is related to the represetation of a category $C_i$.

### *Definition: Category Profile*

A word $w$ has a weight $\Omega(w, C_i)$ in a document category vector $\underline{C}_i$ defined as:

$$\Omega(w, C_i) = \Omega^i_w = max\left\{0, \frac{\beta}{|T_i|}\sum_{d \in T_i} \omega^d_w - \frac{\gamma}{|\overline{T_i}|}\sum_{d \in \overline{T_i}} \omega^d_w\right\}$$

where $T_i$ is the set of training documents classified in $C_i$ and $\overline{T_i}$ are the set of training document not classified in $C_i$

# Rocchio: document and category vectors

Document and Category vectors are derived from the weights assigned to all the words in the vocabulary of a given collection.
A word is added to the vocabulary $V$ whenever it appears in at least one document, altough several feature selection methods can be applied.

**Category Profile, $\underline{C}_i$**

$$\underline{C}_i = \begin{pmatrix} \Omega_1^i \\ . \\ . \\ . \\ \Omega_M^i \end{pmatrix}$$

# Rocchio: document and category vectors

Document and Category vectors are derived from the weights assigned to all the words in the vocabulary of a given collection.
A word is added to the vocabulary $V$ whenever it appears in at least one document, altough several feature selection methods can be applied.

**Category Profile, $\underline{C}_i$**

$$\underline{C}_i = \begin{pmatrix} \Omega_1^i \\ . \\ . \\ . \\ \Omega_M^i \end{pmatrix}$$

# Rocchio: document and category vectors

Document and Category vectors are derived from the weights assigned to all the words in the vocabulary of a given collection.
A word is added to the vocabulary $V$ whenever it appears in at least one document, altough several feature selection methods can be applied.

**Category Profile, $\underline{C}_i$**

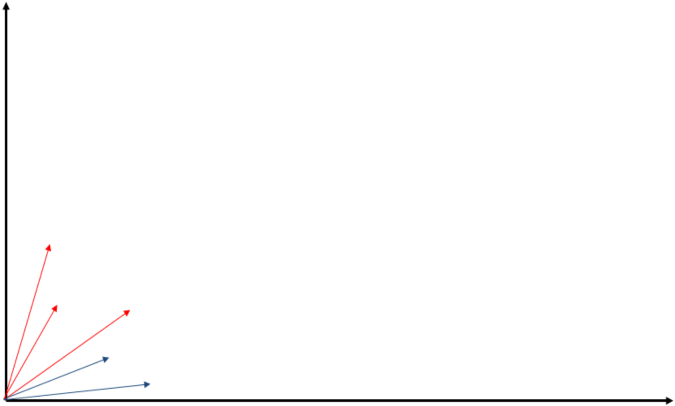
$$\underline{C}_i = \begin{pmatrix} \Omega_1^i \\ \cdot \\ \cdot \\ \cdot \\ \Omega_M^i \end{pmatrix}$$

**Document Vector, $\underline{d}$**

$$\underline{d} = \begin{pmatrix} \omega_1^d \\ \cdot \\ \cdot \\ \cdot \\ \omega_M^d \end{pmatrix}$$

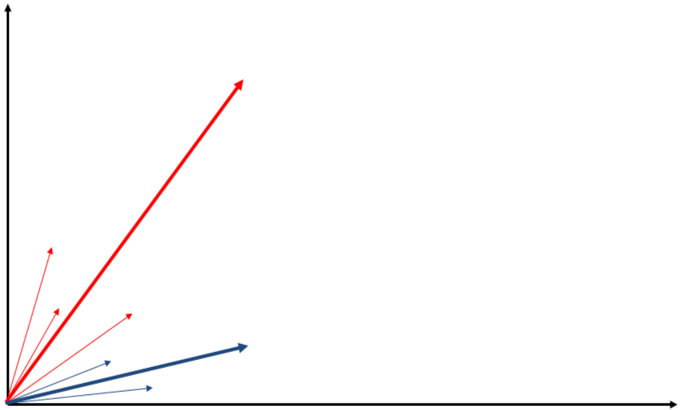
# *Bidimensional View of Rocchio: training set*

Given two classes of training vectors, red and blue instances:



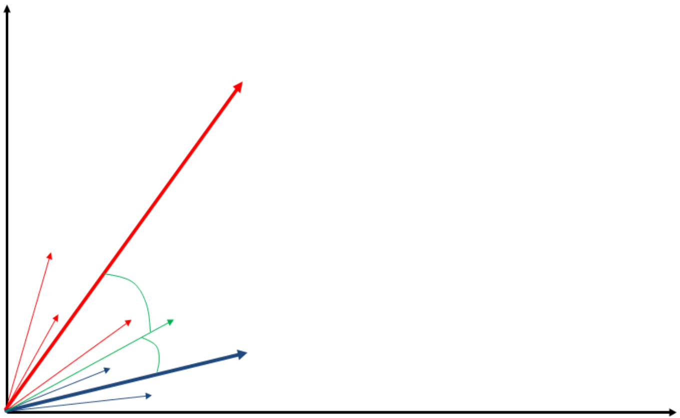
# Bidimensional View of Rocchio: training

Category profiles describe the average behaviour of one class:



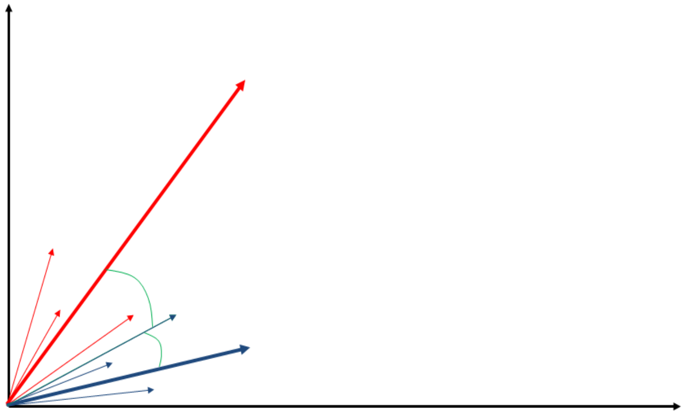
# Bidimensional View of Rocchio: novel input instances

The cosine distances with the new input instance $\underline{d}$ are inversely proportional to the size of the angle between $\underline{C}_i$ and $ud$:



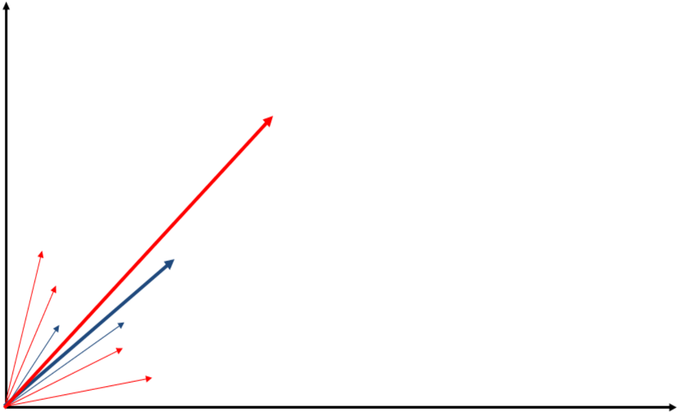
# *Bidimensional View of Rocchio: classifying*

As $(\underline{d}, \underline{C}_{red}) < (\underline{d}, \underline{C}_{blue})$ the new document $d$ is lastly classified in the class of blue instances.



# Limitation of the Rocchio: polymorphism

Prototype-based models have problems with polymorphic (i.e. disjunctive) categories.



# *Memory-based Learning*

Memory-based learning: learning is just storing the representations of the training examples in the collection $T$.

## *Overview of MBL*

The task is again:

- Testing instance $x$:
- Compute similarity between $x$ and all examples in $D$.
- Assign $x$ the **category of the most similar examples in** $D$.

# Memory-based Learning

Memory-based learning: learning is just storing the representations of the training examples in the collection $T$.

## Overview of MBL

The task is again:

- Testing instance $x$:
- Compute similarity between $x$ and all examples in $D$.
- Assign $x$ the **category of the most similar examples in** $D$.

Does not explicitly compute a generalization or category prototypes.

# *Memory-based Learning*

Memory-based learning: learning is just storing the representations of the training examples in the collection $T$.

## *Overview of MBL*

The task is again:

- Testing instance $x$:
- Compute similarity between $x$ and all examples in $D$.
- Assign $x$ the **category of the most similar examples in** $D$.

Does not explicitly compute a generalization or category prototypes.

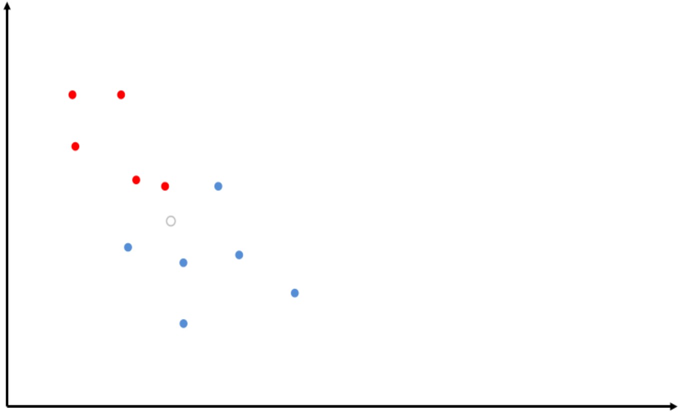
## *Variants of MBL*

The general perspective of MBL is also called:

- Case-based (reasoning as retrieval of most similar cases)
- Memory-based (*memory* as examples are stored for later use)
- Lazy learning (*Lazy* as no model is built, so no generalization is attempted)

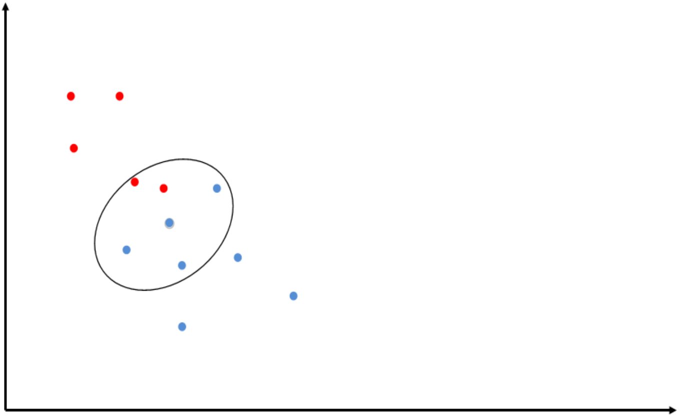
# MBL as Nearest Neighborough Voting

Labeled instances provides a rich description of a newly incoming instance within the space region close enogh to the new example.



# *k-NN classification (k=5)*

Whenever only the $k$ instances closest to the example are used the $k$-NN algorithm is obtained through the voting across $k$ labeled instances.



# $k$-NN: the algorithm

For each each training example $< x, c(x) > \in D$
Compute the corresponding TF-IDF vector, $\underline{x}$, for document $x$.

Test instance $y$:
Compute TF-IDF vector $\underline{y}$ for document $y$.
For each $< x, c(x) > \in D$

$$s_x = cosSim(\underline{y}, \underline{x}) = \frac{(\underline{y}, \underline{x})}{||\underline{x}|| \cdot ||\underline{y}||}$$

Sort examples $x \in D$ by decreasing values of $s_x$.
Let $kNN$ be the set of the closest (i.e. first) $k$ examples in $D$.

RETURN the majority class of examples in $kNN$.

# Similarity

## *The role of similarity among vectors*

In most of the examples above, document data are espressed as high-dimensional vectors, characterized by very sparse term-by-document matrices with positive ordinal attribute values and a significant amount of outliers.

# Similarity

## *The role of similarity among vectors*

In most of the examples above, document data are espressed as high-dimensional vectors, characterized by very sparse term-by-document matrices with positive ordinal attribute values and a significant amount of outliers. In such situations, one is truly faced with the ‘curse of dimensionality’ issue since, even after feature reduction, one is left with **hundreds of dimensions** per object.

# *Similarity and dimensionality reduction*

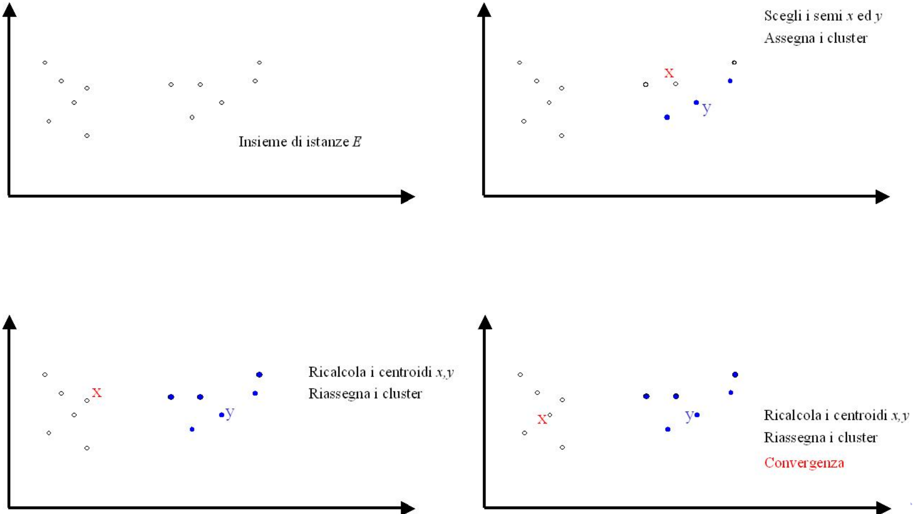
Clustering can be applied to documents to redce the dimensions to take into account. Key cluster analysis activities can be thus devised:

## *Clustering steps*

- *Representation of raw objects* (i.e. documents) into *vectors* of properties with real-valued scores (term weights)
- Definition of a *proximity measure*
- Clustering algorithm
- Evaluation

# Similarity and Clustering

Clustering is a complex process as it requires a search within the set of all possible subsets. A well-known example of clustering algorithm is $k$-mean.



# Similarity

## Clustering steps

- To obtain features $\mathbf{X} \in \mathscr{F}$ from the raw objects, a suitable object representation has to be found.
- Given an objext $O \in \mathscr{D}$, we will refer to such a representation as the feature vector $\underline{x}$ of $X$.
- In the second step, a measure of proximity $\mathbf{S} \in \mathscr{S}$ has to be defined between objects, i.e. $\mathbf{S} : \mathscr{D}^2 \rightarrow \mathbb{R}$. **The choice of similarity or distance can have a deep impact on clustering quality**.

# Minkowski distances

## Minkowski distances

The *Minkowski distances* $L_p(\underline{x}, \underline{y})$ defined as:

$$L_p(\underline{x}, \underline{y}) = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p}$$

are the standard metrics for geometrical problems.

## Euclidean Distance

For $p = 2$ we obtain the Euclidean distance, $d(\underline{x}, \underline{y}) = \|\underline{x} - \underline{y}\|_2^2$.

# *Minkowski distances*

There are several possibilities for converting an $L_p(\underline{x}, \underline{y})$ distance metric (in $[0, \infty)$, with 0 closest) into a *similarity measure* (in $[0, 1]$, with 1 closest) by a monotonic decreasing function.

### *Relation between distances and similarities*

For Euclidean space, we chose to relate distances $d$ and similarities $s$ using

$$s = e^{-d^2}$$

Consequently, the *Euclidean* [0,1]-*normalized similarity* is defined as:

$$s^{(\mathrm{E})}(\underline{x}, \underline{y}) = e^{-\|\underline{x}-\underline{y}\|_2^2}$$

# Similarity: discussion

## Scale and Translation invariance

Euclidean similarity is *translation invariant* ...
but *scale sensitive* while cosine is *translation sensitive* but *scale invariant*.
The extended Jaccard has aspects of both properties as illustrated in figure.
Iso-similarity lines at $s = 0.25$, $0.5$ and $0.75$ for points $\underline{x} = (3,1)^T$ and $\underline{y} = (1,2)^T$ are shown for Euclidean, cosine, and the extended Jaccard.

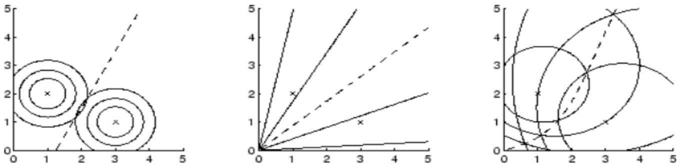



**Figure 4.1:** Properties of (a) Euclidean-based, (b) cosine, and (c) extended Jaccard similarity measures illustrated in 2 dimensions. Two points $(1,2)^\dagger$ and $(3,1)^\dagger$ are marked with $\times$s. For each point iso-similarity surfaces for $s = 0.25$, 0.5, and 0.75 are shown with solid lines. The surface that is equi-similar to the two points is marked with a dashed line.

# *Distance/similarity functions that have not a geometrical origin.*

### *The role of probability*

Very often objects in machine learning are described statistically, i.e. through the notion of distribution of probability that characterizes them: it serves to establish expectations about the values assumed by the object properties (e.g. how likely is 20 as the *age* of the instance of a "*young person*").

Distances are this required to account for the likelihood that a value (e.g. 20) has with respect to others, and amplify (or decrease) the estimates according to such trends: this implies that non linear operators may arise and euclidean distances are not enough. Probability Theory and Information theory thus play a role in establishing some metrics that are useful in some Machine Learning tasks.

# *Distance/similarity functions that have not a geometrical origin.*

### *Other evidence*

Other evidences also stem from extensions of the notion of standard set, such as the fuzzy sets. Fussy sets are usually characterized by smoothed membership functions that range not in the crisp set of $\{0, 1\}$ but in the full range of $[0, 1]$ real values.
In this cases, some definitions emerge from similarity operators deriving from standard set theory, such as the Dice and Jaccard measures.

# *Pearson Correlation*

In collaborative filtering, correlation is often used to predict the specific property of an object, e.g. $\underline{x}$, from a highly similar mentor group for objects, e.g. $\underline{y}$, whose features are known. The result is that an analogy between $\underline{x}$ and $\underline{y}$ is based on the equivalent judgments that mentors $m_1, \ldots., m_n$ provide about both objects.

## *Pearson Correlation*

We need to measure the *analogy between objects* $x$ and $y$ as the *correlation between the vectors* $\underline{x}$ and $\underline{y}$, given that pairwise components $x_i$ and $y_i$ are features stemming from equivalent mentors.
The [0,1]-*normalized Pearson correlation*, $s^{(P)}$, is based on the estimate of such correlations as a function of

- the pairwise judgments $x_i$ and $y_i$ of individual mentors
- the average judgment score $\mu_x$ or $\mu_y$ across all mentors.

# *Pearson Correlation:*
# *objects ($\underline{x}$,$\underline{y}$), mentors ($m_i$) and features ($x_i$,$y_i$)*

Vectors $\underline{x}$ and $\underline{y}$ are derived from mentor judgments as follows.

$$m_1, \quad \ldots, \quad m_n$$

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix} \qquad \begin{bmatrix} y_1 \\ y_2 \\ \ldots \\ y_n \end{bmatrix} = \underline{y}$$

| Summary Features | | | |
|---|---|---|---|
| | $\mu_x$ | *valor medio* | $\mu_y$ |
| | $\sigma_x$ | *deviazione standard* | $\sigma_y$ |

As a consequence, summary features are other useful descriptors of collective attitudes of mentors towards objects $x$ and $y$.

# Pearson Correlation

### Pearson Correlation (2)

The [0,1]-*normalized Pearson correlation*, $s^{(\mathrm{P})}$, is defined as:

$$s^{(\mathrm{P})}(\underline{x},\underline{y}) \triangleq \frac{1}{2}\left(\frac{(\underline{x}-\underline{\mu_x})^T(\underline{y}-\underline{\mu_y})}{\|\underline{x}-\underline{\mu_x}\|_2 \cdot \|\underline{y}-\underline{\mu_y}\|_2}+1\right),$$

where $\underline{\mu_c}$ denotes the vector whose all components correspond to the average feature value $\mu_x$ of $\underline{x}$, across all dimensions.

# Pearson Correlation

## *Normalized Pearson Correlation*

The [0,1]-*normalized Pearson correlation* can also be seen as a probabilistic measure as in:

$$ns^{(\mathrm{P})}(\underline{x},\underline{y}) \triangleq r_{xy} \triangleq \frac{\sum x_i y_i - n\mu_x\mu_y}{\sqrt{(\sum x_i^2 - n{\mu_x}^2)}\ \sqrt{(\sum y_i^2 - n{\mu_y}^2)}}$$

$$= \frac{\sum(x_i-\mu_x)(y_i-\mu_y)}{(n-1)\sigma_x\sigma_y},$$

where $\mu_y$ denotes the average feature value of $\underline{x}$ over all dimensions, and $\sigma_x$ and $\sigma_y$ are the standard deviations of $\underline{x}$ and $\underline{y}$, respectively.

# *Normalized Pearson Correlation*

The [0,1]-*normalized Pearson correlation*:

$$r_{xy} \triangleq \frac{\sum(x_i - \mu_x)(y_i - \mu_y)}{(n-1)\sigma_x \sigma_y}$$

is defined only if both of the standard deviations are finite and both of them are nonzero. It is a corollary of the Cauchy-Schwarz inequality that the correlation cannot exceed 1 in absolute value.
The correlation is 1 in the case of an increasing linear relationship, -1 in the case of a decreasing linear relationship, and some value in between in all other cases, indicating the degree of linear dependence between the variables.

# Jaccard Similarity

## Binary Jaccard Similarity

The *binary Jaccard coefficient* measures the degree of overlap between two sets and is computed as the ratio of the number of shared features of $\underline{x}$ AND $\underline{y}$ to the number possessed by $\underline{x}$ OR $\underline{y}$.

## Example

For example, given two sets' binary indicator vectors $\underline{x} = (0,1,1,0)^T$ and $\underline{y} = (1,1,0,0)^T$, the cardinality of their intersect is 1 and the cardinality of their union is 3, rendering their Jaccard coefficient 1/3.

The binary Jaccard coefficient it is often used in retail market-basket applications.

# Extended Jaccard Similarity

## Extended Jaccard Similarity

The *extended Jaccard coefficient* is the generalized notion of the binary case and it is computed as:

$$s^{(\mathrm{J})}(\underline{x}, \underline{y}) = \frac{\underline{x}^T \underline{y}}{\|\underline{x}\|_2^2 + \|\underline{y}\|_2^2 - \underline{x}^T \underline{y}}$$

# Dice coefficient

## Dice coefficient

Another similarity measure highly related to the extended Jaccard is the *Dice coefficient*:

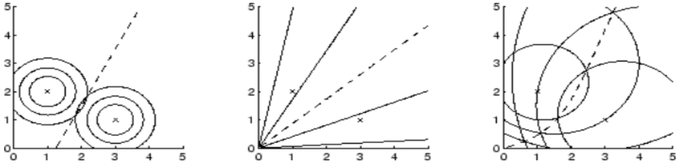
$$s^{(\mathrm{D})}(\underline{x}, \underline{y}) = \frac{2\underline{x}^T \underline{y}}{\|\underline{x}\|_2^2 + \|\underline{y}\|_2^2}$$

The Dice coefficient can be obtained from the extended Jaccard coefficient by adding $\underline{x}^T \underline{y}$ to both the numerator and denominator.

# *Similarity: discussion*

## *Scale and Translation invariance*

Euclidean similarity is *translation invariant* ... but *scale sensitive* while cosine is *translation sensitive* but *scale invariant*. The extended Jaccard has aspects of both properties as illustrated in figure. Iso-similarity lines at $s = 0.25, 0.5$ and $0.75$ for points $\underline{x} = (3, 1)^T$ and $\underline{y} = (1, 2)^T$ are shown for Euclidean, cosine, and the extended Jaccard.



(a) (b) (c)

**Figure 4.1:** Properties of (a) Euclidean-based, (b) cosine, and (c) extended Jaccard similarity measures illustrated in 2 dimensions. Two points $(1, 2)^\dagger$ and $(3, 1)^\dagger$ are marked with $\times$s. For each point iso-similarity surfaces for $s = 0.25$, 0.5, and 0.75 are shown with solid lines. The surface that is equi-similar to the two points is marked with a dashed line.

# Similarity: discussion

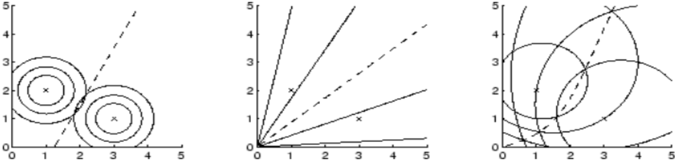



**Figure 4.1:** Properties of (a) Euclidean-based, (b) cosine, and (c) extended Jaccard similarity measures illustrated in 2 dimensions. Two points $(1,2)^\dagger$ and $(3,1)^\dagger$ are marked with $\times$s. For each point iso-similarity surfaces for $s = 0.25$, 0.5, and 0.75 are shown with solid lines. The surface that is equi-similar to the two points is marked with a dashed line.

Thus, for $s^{(J)} \to 0$, extended Jaccard behaves like the cosine measure, and for $s^{(J)} \to 1$, it behaves like the Euclidean distance

# *Similarity: discussion*

## *Similarity in Clustering*

In traditional Euclidean *k*-means clustering the optimal cluster representative $\mathbf{c}_\ell$ minimizes the sum of squared error criterion, i.e.,

$$\mathbf{c}_\ell = \arg\min_{\bar{z} \in \mathscr{F}} \sum_{\underline{x}_j \in \mathscr{C}_\ell} \|\underline{x}_j - \bar{z}\|_2^2$$

Any convex distance-based objective can be translated and extended to the similarity space.

# Similarity: discussion

## Swtiching from distances to similarity

Consider the generalized objective function $f(\mathscr{C}_\ell, \bar{z})$ given a cluster $\mathscr{C}_\ell$ and a representative $\bar{z}$:

$$f(\mathscr{C}_\ell, \bar{z}) = \sum_{\underline{x}_j \in \mathscr{C}_\ell} d(\underline{x}_j, \bar{z})^2 = \|\underline{x} - \bar{z}\|_2^2.$$

We use the transformation $s = e^{-d^2}$ to express the objective in terms of similarity rather than distance:

$$f(\mathscr{C}_\ell, \bar{z}) = \sum_{\underline{x}_j \in \mathscr{C}_\ell} -\log(s(\underline{x}_j, \bar{z}))$$

# Similarity: discussion

## Switching from distances to similarity

Finally, we simplify and transform the objective using a strictly monotonic decreasing function. Instead of minimizing $f(\mathscr{C}_\ell, \bar{z})$, we maximize

$$f'(\mathscr{C}_\ell, \bar{z}) = e^{-f(\mathscr{C}_\ell, \bar{z})}$$

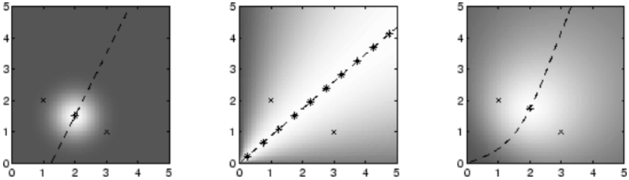
Thus, in the similarity space, the least squared error representative $\mathbf{c}_\ell \in \mathscr{F}$ for a cluster $\mathscr{C}_\ell$ satisfies:

$$\mathbf{c}_\ell = \arg\max_{\bar{z} \in \mathscr{F}} \prod_{\underline{x}_j \in \mathscr{C}_\ell} s(\underline{x}_j, \bar{z})$$

Using the concave evaluation function $f'$, we can obtain optimal representatives for non-Euclidean similarity spaces $\mathscr{S}$.

# Similarity: discussion

To illustrate the values of the evaluation function $f'(\{\mathbf{x}_1, \mathbf{x}_2\}, \mathbf{z})$ are used to shade the background in the figure below.




(a) (b) (c)

**Figure 4.2:** More similarity properties shown on the 2-dimensional example of figure 4.1. The goodness of a location as the common representative of the two points is indicated with brightness. The best representative is marked with a $\star$. The extended Jaccard (c) adopts the middle ground between Euclidean (a) and cosine-based similarity (b).

The maximum likelihood representative of $\underline{x}_1$ and $\underline{x}_2$ is marked with a $\star$.

# Similarity: discussion

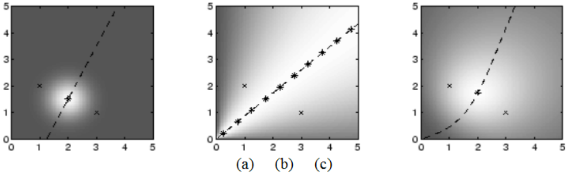



**Figure 4.2:** More similarity properties shown on the 2-dimensional example of figure 4.1. The goodness of a location as the common representative of the two points is indicated with brightness. The best representative is marked with a $\star$. The extended Jaccard (c) adopts the middle ground between Euclidean (a) and cosine-based similarity (b).

For cosine similarity all points on the equi-similarity are optimal representatives. In a maximum likelihood interpretation, we constructed the distance similarity transformation such that

$$p(\bar{z}|\mathbf{c}_\ell) \sim s(\bar{z}, \mathbf{c}_\ell)$$

Consequently, we can use the dual interpretations of probabilities in similarity space $\mathscr{S}$ and errors in distance space $\mathbb{R}$.

# *Information Theory*

Let $\xi$ be a discrete stochastic variable with a finite range $\Omega_\xi = \{x_1, \ldots, x_M\}$ and let $p_i = p(x_i)$ be the corresponding probabilities.

How much information is there in knowing the outcome of $\xi$?

Or equivalently:

How much uncertainty arises if the outcome $\xi$ is unknown?

This is the information needed to specify which of the $x_i$ has occurred. The problem is writing $\xi$.
Let us assume further that we only have a small set of symbols $A = \{a_k : k = 1, \ldots D\}$, that is a *coding alphabet*.

# Entropy

## Uncertainty of $\xi$

The uncertainty introduced by the random variable $\xi$ will be taken to be the *expectation value of the number of digits required to specify its outcome*. This is the expectation value of $-\log_2 P(\xi)$, i.e.

$$E[-\log_2 P(\xi)] = \sum_i -p_i \log_2 p_i$$

# Entropy

## Entropy

The entropy $H[\xi]$ of $\xi$ is precisely the amount of uncertainty introduced by the random variable $\xi$ and it is more often referred to a natural logarithm $\ln(.)$, so that

$$H[\xi] = E[-\ln p(\xi)] = \sum_{x_i \in \Omega_\xi} -p(x_i)\ln p(x_i) = \sum_{i}^{M} -p_i \ln p_i$$

# Entropy

### Example 1: Rolling the dice

In the Die example, $\forall i = 1, ..., 6$, it follows that $p_i = \frac{1}{6}$.

$$H[\xi] = E[-\ln p(\xi)] = \sum_{x_i \in \Omega_\xi} -p(x_i) \ln p(x_i) = 6 \cdot \frac{1}{6} \ln 6 = 1,792$$

### Example 2: A loosing Die

A loosing Die: $p_1 = 1.00,$ and $\forall i = 2, ..., 6$, $p_i = 0$.

$$H[\xi] = E[-\ln p(\xi)] = \sum_{x_i \in \Omega_\xi} -p(x_i) \ln p(x_i) = 1 \ln 1 = 0$$

# Entropy

### Consequence

Given a distribution $p_i \quad (i = 1, , \ldots, M)$ for a discrete random variable $\xi$ then for any other distribution $q_i \quad (i = 1, , \ldots, M)$ over the same sample space $\Omega_\xi$ it follows that:

$$H[\xi] = -\sum_i^M p_i \ln p_i \leq -\sum_i^M p_i \ln q_i$$

where equality holds **iff** the two distribution are the same, i.e. $\forall i = 1, \ldots, M \qquad p_i = q_i$

# *Joint-Entropy*

Given two random variable $\xi$ and $\eta$:

## *Joint-Entropy*

the *joint entropy* of $\xi$ and $\eta$ is defined as:

$$H[\xi,\eta] = -\sum_{i=1}^{M}\sum_{j=1}^{L} p(x_i,y_j)\ln p(x_i,y_j) = H[\eta,\xi]$$

# Conditional-entropy

## *Conditional Entropy*

the *conditional entropy* $H[\xi|\eta]$ of $\xi$ and $\eta$ is defined as:

$$
\begin{aligned}
H[\xi|\eta] &= -\sum_{j=1}^{L} p(y_j) \sum_{i=1}^{M} p(x_i|y_j) \ln p(x_i|y_j) = \\
&= -\sum_{j=1}^{L} \sum_{i=1}^{M} p(x_i, y_j) \ln p(x_i|y_j)
\end{aligned}
$$

# Conditional and joint entropy

## Conditional and Joint Entropy

The conditional and joint entropies are related just like the conditional and joint probabilities:

$$H[\xi, \eta] = H[\eta] + H[\xi|\eta]$$

## Conveyed Information

The *information conveyed* by $\eta$, denoted $I[\xi|\eta]$, is the reduction in entropy of $\xi$ by finding out the outcome of $\eta$. This is defined by:

$$I[\xi|\eta] = H[\xi] - H[\xi|\eta]$$

# Mutual Information

Given two random variable $\xi$ and $\eta$:

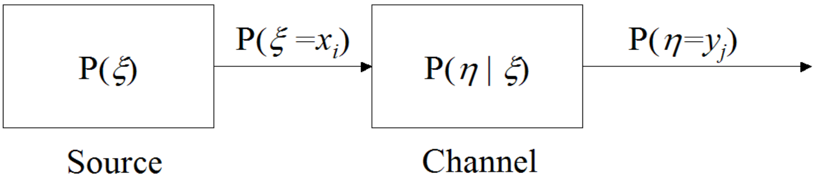
## Mutual Information

The *mutual information* between $\xi$ and $\eta$ is defined as:

$$
\begin{aligned}
MI[\xi,\eta] &= E[\ln \frac{P(\xi,\eta)}{P(\xi)\cdot P(\eta)}] = \\
&= \sum_{(x,y)\in\Omega_{(\xi,\eta)}} f_{(\xi,\eta)}(x,y) \ln \frac{f_{(\xi,\eta)}(x,y)}{f_\xi(x)\cdot f_\eta(y)}
\end{aligned}
$$

# *Mutual Information*

Mutual Information measures the amount of information about a random variable $\xi$ an observer receives when the outcome of a random variable $\eta$ is available.




How much information about the source output $x_i$ does an observer gain by knowing the channel output $y_j$?

# Mutual Information

Mutual Information measures the amount of information about a random variable $\xi$ an observer receives when the outcome of a random variable $\eta$ is known, in fact:

## Mutual Information

$$
\begin{aligned}
MI[\xi,\eta] &= H[\xi] - H[\xi|\eta] = \\
&= \sum_{(x,y)\in\Omega_{(\xi,\eta)}} f_{(\xi,\eta)}(x,y) \ln \frac{f_{(\xi,\eta)}(x,y)}{f_{\xi}(x)\cdot f_{\eta}(y)}
\end{aligned}
$$

# Pointwise Mutual Information

Another way to look to mutual information is about the individual values (i.e. outcomes) $\xi = x_i$ and $\eta = y_j$.

## Pointwise Mutual Information

Given the two random variable $\xi$ and $\eta$: the *pointwise mutual information* between $\xi = x_i$ and $\eta = y_j$ is defined as:

$$MI[x_i, y_j] = f_{(\xi,\eta)}(x_i, y_j) \ln \frac{f_{(\xi,\eta)}(x_i, y_j)}{f_\xi(x_i) \cdot f_\eta(y_j)} = P(x_i, y_j) \ln \frac{P(x_i, y_j)}{P(x_i) \cdot P(y_j)}$$

# Pointwise Mutual Information

## Pointwise Mutual Information (pmi)

$$MI[x_i, y_j] = P(x_i, y_j) \ln \frac{P(x_i, y_j)}{P(x_i) \cdot P(y_j)}$$

## Use of the pmi

If $MI[x_i, y_j] >> 0$, there is a strong correlation between $x_i$ and $y_j$
If $MI[x_i, y_j] << 0$, there is a strong negative correlation.
When $MI[x_i, y_j] \approx 0$ the two outcomes are almost independent.

# Cross-entropy

## Cross-entropy

If we have two distributions (collections of probabilities) $p(x)$ and $q(x)$ on $\Omega_\xi$, then the *cross entropy* of $q$ with respect to $p$ is given by:

$$H_p[q] = -\sum_{x\in\Omega_\xi} p(x)\ln q(x)$$

## Minimality

$$H_p[q] = -\sum_{x\in\Omega_\xi} p(x)\ln q(x) \geq -\sum_{x\in\Omega_\xi} p(x)\ln p(x) \quad \forall q$$

implies that the cross entropy of a distribution $q$ w.r.t. another distribution $p$ is **minimal** when $q$ is identical to $p$.

# *Cross-entropy as a Norm*

## *Cross-entropy*

$$H_p[q] = -\sum_{x \in \Omega_\xi} p(x) \ln q(x)$$

## *Relative Entropy (or Kullback-Leibler distance)*

$$D[p||q] = \sum_{x \in \Omega_\xi} p(x) \ln \frac{p(x)}{q(x)} = H_p[q] - H[p]$$

# Cross-entropy and Norms

## Relative Entropy (or Kullback-Leibler distance)

$$D[p||q] = \sum_{x \in \Omega_\xi} p(x) \ln \frac{p(x)}{q(x)} = H_p[q] - H[p]$$

## KL distance: properties

$$D[p||q] \geq 0 \quad \forall q$$

$$D[p||q] = 0 \qquad \textbf{iff } q = p$$

# *Cross-entropy and Norms*

### *Relative Entropy (or Kullback-Leibler distance)*

$$D[p||q] = \sum_{x \in \Omega_\xi} p(x) \ln \frac{p(x)}{q(x)} = H_p[q] - H[p]$$

### *KL distance as a norm?*

Unfortunately, as

$$D[p||q] \neq D[q||p]$$

the KL distance is *not* a valid metric in the classical terms. It is a *measure of the dissimilarity* between $p$ and $q$.

# *Norms, Similarity and Learning*

### *Why ranking probability distributions is necessary?*

- During a learning process we need to figure out the circumstances (i.e. the state of affairs of the world) under which a certain concept/class/property manifest.
- This make a direct reference to the probability of some (stochastic) event. Stochastic events are used to describe circumstances and properties.
- Moreover, learning proceeds from experience, i.e. known facts or previous classified examples, to rules, i.e. probability joint distributions over *decisions* and *circumstances*
- Learning in general means **to induce the proper probability distributions from the known examples**. There are several many ways to do it!!!

# *Norms, Similarity and Learning*

### *Why ranking probability distributions is necessary?*

- **Consequences.** In general, we need to compare different inductive hypothesis (*IH*), that are different probability distributions $q_i$ of the same decision,
- In order to do it, we measure the agreement of our hypothesis with the observations (i.e. a pool of annotated data kept aside, the *held out*, to validate the different $q_i$)
- The result is an estimate of the similarity between the probability $q_i$ induced at the $i$-th learning stage with the probability $p$ characterizing the known examples.
- The KL divergence $D[p||q] = H_p(q) - H(p)$ can be the suitable dissimilarity function.
- The probability $\hat{q}$ (such that $\hat{q}$ minimizes $\forall i \quad D[p||q_i]$) is returned.

# Further similarity measures

## Vector similarities

- Grefenstette (fuzzy) set-oriented similarity for capturing dependency relations (head words)

## Distributional (Probabilstic) similarities

- Lin similarity (commonalities) (Dice like)

$$sim(\underline{x},\underline{y}) = \frac{2 \cdot \log P(common\_dep(\underline{x},\underline{y}))}{\log P(\underline{x}) + \log P(\underline{y})}$$

- *Jensen-Shannon* total divergence to the mean:

$$A(p,q) = D(p \| \frac{p+q}{2}) + D(q \| \frac{p+q}{2})$$

- $\alpha$-skewed divergence (Lee, 1999): $s_\alpha(p,q) = D(p \| \alpha p + (1-\alpha)q)$ ($\alpha = 0,1$ or $0.01$)

# *Vector Space Modeling References*

### *Vectors, Operations, Norms and Distances*

K. Van Rijesbergen, The Geometry of Information Retrieval, CUP Press, 2004.

### *Distances and Similarities*

Alexander Strehl, Relationship-based Clustering and Cluster Ensembles for High-dimensional Data Mining, PhD Dissertation, University of Texas at Austin, 2002. URL: `http://www.lans.ece.utexas.edu/~strehl/diss/htdi.html`.

### *Nice collection of code and definitions*

Sam- string metrics. URL: `http://www.dcs.shef.ac.uk/~sam/stringmetrics.html`.

# *Probability and Information References*

## *Elementary Information Theory*

- in (Krenn & Samuelsson, 1997), Brigitte Krenn, Christer Samuelsson, *The Linguist's Guide to Statistics Don't Panic*, Univ. of Saarlandes, 1997.
  URL: `http://nlp.stanford.edu/fsnlp/dontpanic.pdf`