Knowledge Discovery in Databases II

Winter Term 2015/2016

Lecture 2: High-Dimensional Feature Vectors

Lectures : Prof. Dr. Peer Kröger, Yifeng Lu
Tutorials: Yifeng Lu

Script © 2015, 2017 Eirini Ntoutsi, Matthias Schubert, Arthur Zimek, Peer Kröger, Yifeng Lu

http://www.dbs.ifi.lmu.de/cms/Knowledge_Discovery_in_Databases_II_(KDD_II)

Outline

1. Introduction to Feature Spaces
2. Challenges of high dimensionality
3. Feature Selection
4. Feature Reduction and Metric Learning
5. Clustering in High-Dimensional Data

DATABASE SYSTEMS GROUP

Feature Transform and Similarity Model

LMU

- **Feature Transform**
  - Consider the following spaces:
    - $\mathbb{U}$ denotes the universe of data objects
    - $\mathbb{F} \subseteq \mathbb{R}^n$ denotes an n-dimensional feature space
  - A feature transformation is a mapping $f : \mathbb{U} \rightarrow \mathbb{R}^n$ of objects from $\mathbb{U}$ to the feature space $\mathbb{F}$.

- **Similarity Model**
  - A similarity model S:$\mathbb{U}\times\mathbb{U}\rightarrow \mathbb{R}$ is defined for all objects $p,q \in \mathbb{U}$ as:

$$S(p,q)=sim(f(p),f(q))$$

where

$$sim: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$$

is a similarity measure or a dissimilarity (distance) measure in $\mathbb{F}$.

DATABASE SYSTEMS GROUP

Similarity vs. Dissimilarity

LMU

- **Small but important difference**
  - A ***similarity measure*** (sim) assigns high values to similar objects:

    sim(p,q) $\geq$ sim(p,r)

  - A ***dissimilarity measure*** (δ) assigns low values to similar objects:

    δ(p,q) $\leq$ δ(p,r)

object p

object q

object r

Dissimilarity

- **Dissimilarity measures follow the idea of the geometric approach**
  - objects are defined by their perceptual representations in a perceptual space
  - perceptual space = psychological space
  - geometric distance between the perceptual representations defines the (dis)similarity of objects
- **Within the scope of Feature-based similarity:**
  - perceptual space = feature space $\mathbb{F}$ or feature representation space $\mathbb{R}^n$
  - geometric distance = distance function

Dissimilarity

- **Distance Space**
  - The tuple $(\mathbb{F},\delta)$ is called a distance space if δ is a distance function, i.e. it satisfies reflexivity, non-negativity, and symmetry.
- **Metric Space**
  - The tuple $(\mathbb{F},\delta)$ is called a metric space if δ is a metric function, i.e. it is a distance function (see above) and it satisfies the triangle inequality

DATABASE SYSTEMS GROUP

Dissimilarity

- **Discussion**
  - Sound mathematical interpretation
  - (Metric) distance functions allow domain experts to model their notion of dissimilarity
  - Allow to tune efficiency of data mining approaches (particularly the utilization of the triangle inequality)
  - Powerful and general: independent adaptation/utilization without knowing the inner-workings of a (metric) distance function
  - Long-lasting discussion of whether the distance properties and in particular the metric properties reflect the perceived dissimilarity correctly, see the following contradicting example:

no properties shared alike — similar w.r.t. roundness — similar w.r.t. luminosity

DATABASE SYSTEMS GROUP

Similarity

- **Similarity function**
  - quantifies the similarity between two objects
  - corresponds to the notion that nothing is more similar than the same
  - satisfies the symmetry and maximum self-similarity properties

DATABASE SYSTEMS GROUP

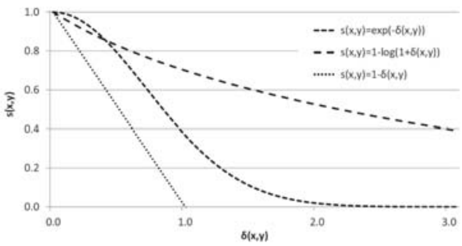
Similarity vs. Dissimilarity

LMU

- **Transformation**
  - Let $\mathbb{F}$ be a feature space and $\delta: \mathbb{F} \times \mathbb{F} \to \mathbb{R}$ be a distance function
  - Any monotonically decreasing function $f: \mathbb{R} \to \mathbb{R}$ defines a similarity function $s: \mathbb{F} \times \mathbb{F} \to \mathbb{R}$ as follows:

$$\forall x,y \in \mathbb{X}: s(x,y) = f(\delta(x,y))$$

- **Some prominent similarity functions ($x,y \in \mathbb{F}$):**
  - exponential: $s(x,y)=e^{(-\delta(x,y))}$
  - logarithmic: $s(x,y)=1-\log(1+\delta(x,y))$
  - linear: $s(x,y)=1-\delta(x,y)$



DATABASE SYSTEMS GROUP

Examples (only very few …)

LMU

- **Similarity ($x,y \in \mathbb{F} \subseteq \mathbb{R}^d$):**
  - Dot-Product $\quad x \cdot y^T = \sum_{i=1}^{d} x_i \cdot y_i = \|x\| \cdot \|y\| \cdot \cos\varphi$
  - Cosine $\quad \frac{x \cdot y^T}{\|x\| \cdot \|y\|} = \frac{\sum_{i=1}^{d} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{d} x_i^2} \cdot \sqrt{\sum_{i=1}^{d} y_i^2}}$
  - Pearson Correlation $\quad \frac{\sum_{i=1}^{d} (x_i - \overline{x_i}) \cdot (y_i - \overline{y_i})}{\sqrt{\sum_{i=1}^{d} (x_i - \overline{x_i})^2} \cdot \sqrt{\sum_{i=1}^{d} (y_i - \overline{y_i})^2}}$
  - Kernels …
- **Distance ($x,y \in \mathbb{F} \subseteq \mathbb{R}^d$):**
  - Lp-norms (aka Minkowski metric) $\quad : \quad L_p(x,y) = \left(\sum_{1 \le i \le d} |x_i - y_i|^p\right)^{\frac{1}{p}}$

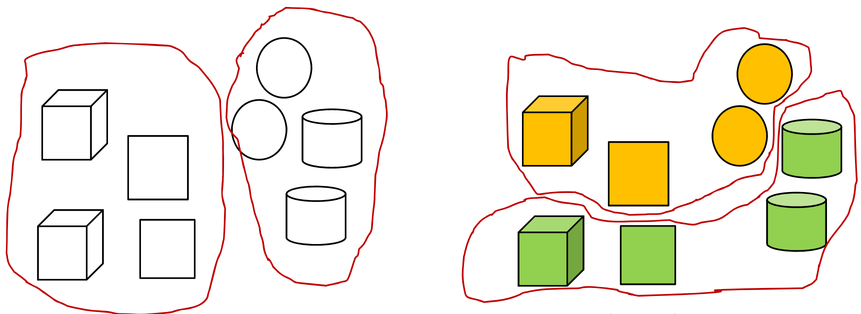
    Fractional Minkowski Dist. ($p < 1$), Manhattan Dist. ($p = 1$), Euclidean Dist. ($p = 2$), Chebyshev/Maximum Dist. ($p = \infty$)
  - Mahalanobis (aka quadratic forms)
  - Hamming: $\quad \text{HammingDist}(x,y) = \sum_{1 \le i \le d} \begin{cases} 1 & \text{if } x_i \ne y_i \\ 0 & \text{else} \end{cases}$
  - …

Outline

1. Introduction to Feature Spaces
2. Challenges of high dimensionality
3. Feature Selection
4. Feature Reduction and Metric Learning
5. Clustering in High-Dimensional Data

The more the merrier or more is less?

- Motivating Example: baby shapes game (truly motivating for students …)



Based on shape grouping   Based on color grouping

What about grouping based on both shape and color?

High-Dimensional Data: why bother?

- **The good old days of data mining …**
  - Data generation and, to some extend, data storage was costly (sic!)
  - Domain experts carefully considered which features/variables to measure before designing the experiment/the feature transform/…
  - Consequence: also data sets were well designed and potentially contained only a small number of relevant features

- **Nowadays, data science is also about integrating everything**
  - Generating and storing data is easy and cheap
  - People tend to measure everything they can and even more (including even more complex feature transformations)
  - The Data Science mantra is often interpreted as “analyze data from as many sources as (technically) possible”
  - Consequence: data sets are high-dimensional containing a large number of features; the relevancy of each feature for the analysis goal is not clear *a priori*

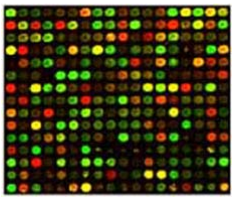
Examples of High-Dimensional Data 1/2

- **Image data**
  - low-level image descriptors
    (color histograms, textures, shape information ...)
  - If each pixel a feature, a 64x64 image → 4,096 features
  - Regional descriptors
  - Between 16 and 1,000 features

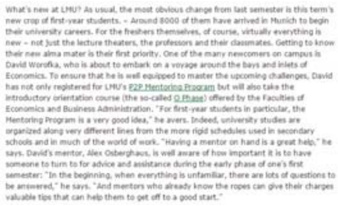
- **Metabolome data**
  - feature = concentration of one metabolite
  - The term metabolite usually restricted to small molecules, that are intermediates and products of metabolism.
  - The Human Metabolome Database contains 41,993 metabolite entries
  - Bavaria newborn screening (For each newborn in Bavaria, the blood concentrations of 43 metabolites are measured in the first 48 hours after birth)
  - between 50 and 2,000 features

Examples of High-Dimensional Data 2/2

- **Microarray data**
  - Features correspond to genes
  - Thousands or tens of thousands of genes in a single experiment
  - Up to 20,000 features
  - Dimensionality is much higher than the sample size



- **Text data**
  - Features correspond to words/terms
  - Different documents have different words
  - between 5,000 and 20,000 features
  - Very often, esp. in social media,
    - Abbreviations (e.g., Dr)
    - colloquial language (e.g., luv)
    - Special words (e.g, hashtags, @TwitterUser)



What's new at LMU? As usual, the most obvious change from last semester is this term's new crop of first-year students. – Around 8000 of them have arrived in Munich to begin their university careers. For the freshers themselves, of course, virtually everything is new – not just the lecture theaters, the professors and their classmates. Getting to know their new alma mater is their first priority. One of the many newcomers on campus is David Worofka, who is about to embark on a voyage around the bays and inlets of Economics. To ensure that he is well equipped to master the upcoming challenges, David has not only registered for LMU's P2P Mentoring Program but will also take the introductory orientation course (the so-called O Phase) offered by the Faculties of Economics and Business Administration. "For first-year students in particular, the Mentoring Program is a very good idea," he avers. Indeed, university studies are organized along very different lines from the more rigid schedules used in secondary schools and in much of the world of work. "Having a mentor on hand is a great help," he says. David's mentor, Alex Osberghaus, is well aware of how important it is to have someone to turn to for advice and assistance during the early phase of one's first semester: "In the beginning, when everything is unfamiliar, there are lots of questions to be answered," he says. "And mentors who already know the ropes can give their charges valuable tips that can help them to get off to a good start."

Excerpt from LMU website: http://tinyurl.com/qhq6byz

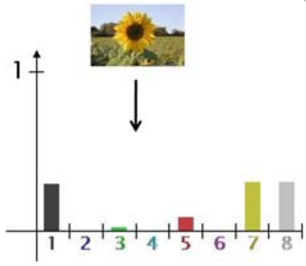
Intrinsic problems of traditional approaches

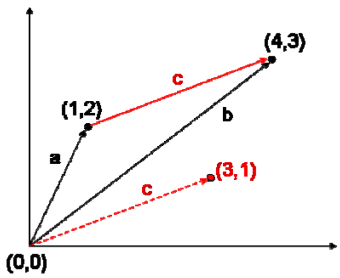
- Data objects (e.g. images) are represented as d-dimensional feature vectors (e.g. color histograms)
- 2-dimensional example:
  - *a* and *b* are 2-dimensional vectors
  - The Euclidean distance between *a* and *b* is:

$$dist_2[(1,2),(4,3)] =$$
$$\sqrt{(1-4)^2+(2-3)^2} = \sqrt{10}$$

and it corresponds to the norm of the difference vector *c*

$$\|c\|_2 = \sqrt{3^2+1^2}$$





Distances grow alike

- **With increasing dimensionality, distances grow, too:**
  - Example: $dist_2[(1,2),(4,3)] = \sqrt{10}$
    double the feature vector length (double the original features)
    $dist_2[(1,2,1,2),(4,3,4,3)] = \sqrt{3^2 + 1^2 + 3^2 + 1^2} = \sqrt{20}$
  - Effect seems not so important, values might be only in a larger scale?

- **Contrast is lost in high dimensional data:**
  - Distances grow *more and more alike*
  - Distances concentrate in small value range (low variance)

  → No clear distinction between clustered objects

Concentration of the Norms and Distances

- ***Concentration phenomenon:***
  As dimensionality grows, the contrast provided by usual metrics decreases. In other words, the distribution of norms in a given distribution of points tends to concentrate

- Example: Euclidean norm of vectors consisting of several variables that are independent and identically distributed :

$$\|y\|_2 = \sqrt{y_1^2 + y_2^2 + \cdots + y_d^2}$$

- In high dimensional spaces this norm behaves unexpectedly

Concentration of the Norms and Distances

Theorem

Let $\boldsymbol{y}$ be a d-dimensional vector $[y_1, \ldots, y_d]$; all components $y_i, 1 \le i \le d$, are independent and identically distributed:

Then the mean and the variance of the Euclidean norm are:

$$\mu_{\|y\|} = \sqrt{ad - b} + \mathcal{O}(d^{-1}) \quad \text{and} \quad \sigma_{\|y\|} = b + \mathcal{O}(d^{-\frac{1}{2}})$$

where a and b are parameters depending only on the central moments of order 1, 2, 3, 4.

→ The norm of random variables grows proportionally to $\sqrt{d}$, but the variance remains constant for sufficiently large d

→ with growing dimensionality, the relative error made by taking $\mu_{\|y\|}$ instead of $\|y\|$ becomes negligible

Neighborhoods become meaningless (part 1)

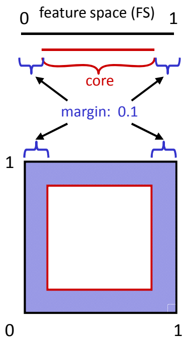
- **Using neighborhoods is based on a key assumption:**
  - Objects that are similar to an object o are in its neighborhood
  - Object that are dissimilar to o are not in its neighborhood
- **What if all objects are in the same neighborhood?**
  - Consider effect on distances: kNN distances are almost equal to each other
  - → k nearest neighbor is a random object

Challenges due to high dimensionality: overview

- **Curve of Dimensionality**
  – Distance to the nearest and the farthest neighbor converge

$$\frac{\text{nearestNeighborDist}}{\text{farthestNeighborDist}} \approx 1$$

  – The likelihood that a data object is located on the margin of the data space exponentially increases with the dimensionality



1D: $P_{FS} = 1^1 = 1$
$P_{core} = 0.8^1 = 0.8$
$P_{margin} = 1 - 0.8 = 0.2$

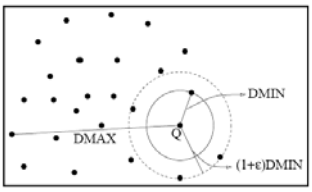
2D: $P_{FS} = 1^2 = 1$
$P_{core} = 0.8^2 = 0.64$
$P_{margin} = 1 - 0.64 = 0.36$

3D: $P_{FS} = 1^3 = 1$
$P_{core} = 0.8^3 = 0.512$
$P_{margin} = 1 - 0.512 = 0.488$

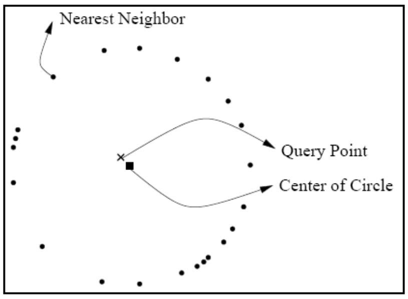
10D: $P_{FS} = 1^{10} = 1$
$P_{core} = 0.8^{10} = 0.107$
$P_{margin} = 1\text{-}0.107\text{=}0.893$

NN Instability Result

Definition:
- A NN-query is *unstable* for a given ϵ if the distance from the query point to most data points is less than $(1 + \epsilon)$ times the distance from the query point to its nearest neighbor.

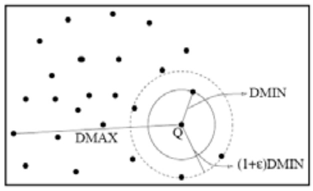


- We will show that with growing dimensionality, the probability that a query is unstable converges to 1



NN Instability Result

- Consider a d-dim. query point Q and N d-dim. sample points $X_1, X_2, \dots, X_N$ (independent and identically distributed)
- We define:
  $DMIN_d = \min\{dist_2(X_i, Q) | 1 \leq i \leq N\}$
  $DMAX_d = \max\{dist_2(X_i, Q) | 1 \leq i \leq N\}$



Theorem: If $\lim_{d\to\infty} \left(\frac{var(dist_2(X_i, Q))}{E[dist_2(X_i, Q)]^2} \right) = 0$

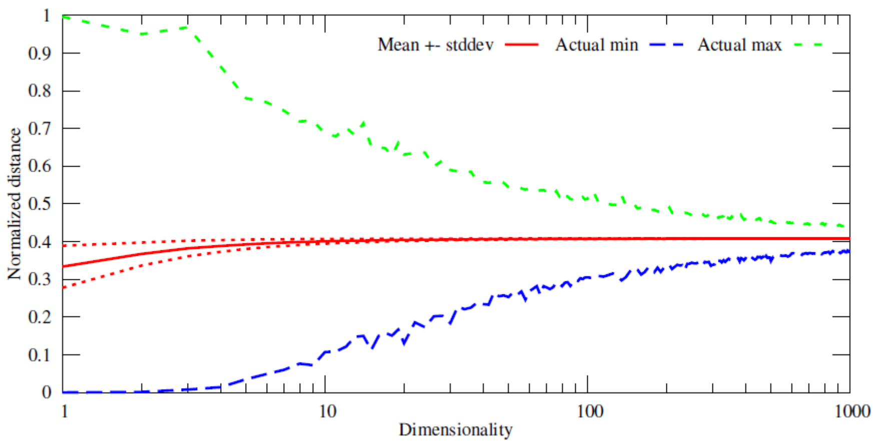
Then $\forall \epsilon > 0 \quad \lim_{d\to\infty} P[DMAX_d \leq (1+\epsilon) DMIN_d] = 1$

If the precondition holds (e.g., if the variance of the distance values remains more or less constant for a sufficiently large d) all points converge to the same distance from the query

→ the concept of the nearest neighbor is no longer meaningful

Challenges due to high dimensionality: distances

- Pairwise distances example: sample of 10^5 instances drawn from a uniform [0, 1] distribution, normalized (1/ sqrt(d)).



Source: Tutorial on Outlier Detection in High-Dimensional Data, Zimek et al, ICDM 2012

Challenges due to high dimensionality: similarity

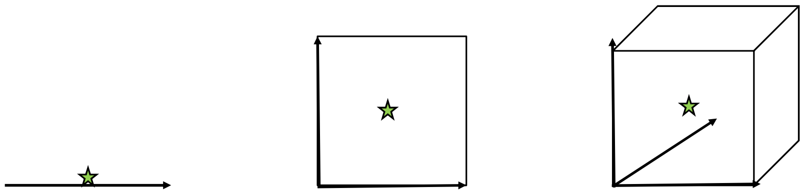
Further explanation of the *Curse of Dimensionality*:

- Consider the feature space of *d relevant* features for a given application
  => truly similar objects display small distances in most features
- Now add *d*x* additional features being *independent* of the initial feature space
- With increasing *x* the distance in the independent subspace will dominate the distance in the complete feature space

⇒ How many relevant features must be similar to indicate object similarity?
⇒ How many relevant features must be dissimilar to indicate dissimilarity?
⇒ With increasing dimensionality the likelihood that two objects are similar in every respect gets smaller.

Challenges due to high dimensionality: hypothesis space

- The more features, the larger the *hypothesis space*

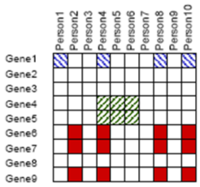


1D 2D 3D

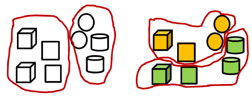
- The lower the hypothesis space
  - the easier to find the correct hypothesis
  - the less examples you need

Challenges due to high dimensionality: this and that

- Patterns and models on high-dimensional data are often *hard to interpret*.
  - e.g., long decision rules
- *Efficiency* in high-dimensional spaces is often limited
  - index structures degenerate
  - distance computations are much more expensive
- Pattern might only be observable in *subspaces* or *projected spaces*



Recall the baby shapes!



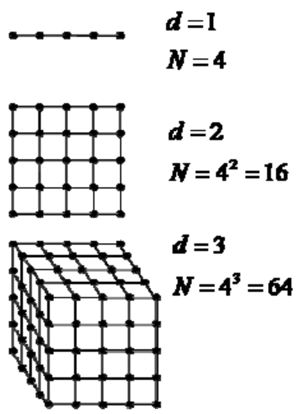
- Cliques of correlated features dominate the object description

Major parts of high dimensional spaces are empty

- In low dimensional spaces we have some (intuitive) assumptions on
  - Behavior of volumes (sphere, cube, etc.)
  - Distribution of data objects
- Basic assumptions do not hold in high dimensional spaces:
  - Space becomes sparse or even empty
    - → Probability of one object inside a fixed range tends to become zero
  - Distribution of data has a strange behavior
    - E.g. a normal distribution has only few objects in its center
    - → Tails of distributions become more important

"The Empty Space Phenomenon"

- Consider a d-dimensional space with partitions of constant size $\frac{1}{m}$
- The number of cells N increases exponentially in d: $N = m^d$
- Suppose x points are randomly placed in this space
- In low-dimensional spaces there are few empty partitions and many points per partitions
- In high-dimensional spaces there are far more partitions than points
  → there are many empty partitions



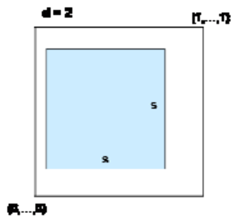
"The Empty Space Phenomenon"

- Consider a simple partitioning scheme, which splits the data in each dimension in 2 halves
- For d dimensions we obtain 2^d partitions
- Consider N = 10^6 samples in this space
- For $d \leq 10$ such a partition makes sense
- For d = 100 there are around 10^{30} partitions, so most partitions are empty

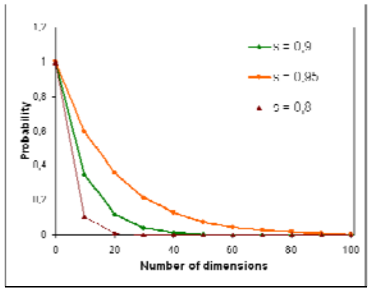
[WSB98] Roger Weber, Hans-Jörg Schek and Stephen Blott: A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In VLDB '98: Proceedings of the 24rd International Conference on Very Large Data Bases.

Data Space is sparsely populated

- Consider a hypercube range query with length s in all dimensions, placed arbitrarily in the data space $[0,1]^d$
- E is the event that an arbitrary point lies within this range query
- The probability for E is $\Pr[E] = s^d$



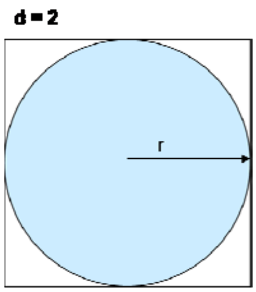
→ with increasing dimensionality, even very large hyper-cube range queries are not likely to contain a point. [WSB98]



Spherical Range Queries

- Consider the largest spherical query that fits entirely within a d-dimensional data space
- Thus for a hypercube with side length $2r$, the sphere has radius r
- E is the event that an arbitrary point lies within this spherical query
- The probability for E is:

$$\Pr[E] = \frac{V_{sphere}(r)}{V_{cube}(r)}$$



- We have:

$$V_{sphere}(r) = \frac{(\sqrt{\pi} \cdot r)^d}{\Gamma(1 + \frac{d}{2})} \qquad V_{cube}(2r) = (2r)^d$$

Spherical Range Queries

- For a growing dimensionality we obtain: $\lim_{d\to\infty} \frac{V_{sphere}(r)}{V_{cube}(2r)} = 0$
- Consider $V_{cube}(2r) = 1$, then $r = 0.5$ and $\lim_{d\to\infty} V_{sphere} = 0$

→ The volume of the sphere vanishes with increasing dimensionality

- The fraction of the volume of the cube contained in the hypersphere is:

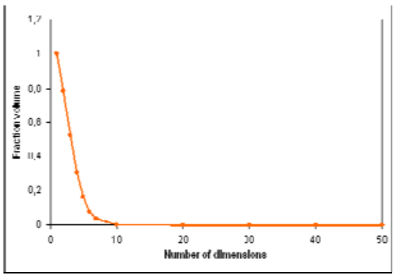
$$f_d = \frac{\sqrt{\pi^d}\, r^d}{\Gamma\left(1+\frac{d}{2}\right)(2r)^d} = \frac{\sqrt{\pi^d}}{\Gamma\left(1+\frac{d}{2}\right)2^d}$$

Dimensionality d	1	2	3	4	5	6	7
Fraction Volume f_d	1	0.785	0.524	0.308	0.164	0.081	0.037

- Since the relative volume of the sphere becomes smaller and smaller, it becomes improbable that any point will be found within this sphere in high dimensional spaces

[WSB98] Roger Weber, Hans-Jörg Schek and Stephen Blott: "A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces". In VLDB '98: Proceedings of the 24rd International Conference on Very Large Data Bases.
[LV07] John A Lee and Michel Verleysen: "Nonlinear Dimensionality Reduction". Springer, 2007.

Sphere Enclosed in Hypercube



- with increasing dimensionality the center of the hypercube becomes less important and the volume concentrates in its corners

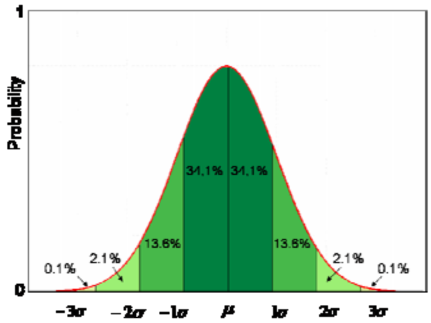
→ distortion of space compared to our 3D way of thinking

Consequence: Importance of the Tails

Intuition for low dimensional data:

- Consider standard density function *f*
- Consider *f'*:

$$f'(x) = \begin{cases} 0, & f(x) < 0.01 \sup f \\ f(x), & else \end{cases}$$



- Rescaling *f'* to a density function will make very little difference in the one dimensional case, since very few data points occur in regions where *f* is very small

Importance of the Tails

For high dimensional data:

- More than half of the data has less then 1/100 of the maximum density *f*(0) (for μ = 0)
- Example: 10-dimensional Gaussian distribution X:

$$\frac{f(X)}{f(0)} = e^{(-\frac{1}{2}X^T X)} \sim e^{(-\frac{1}{2}\chi^2_{10})}$$

since the median of the χ^2_{10} distribution is 9.34,

the median of $\frac{f(X)}{f(0)}$ is $e^{-\frac{9.34}{2}} = 0.0094$

- Thus, most objects occur at the tails of the distribution

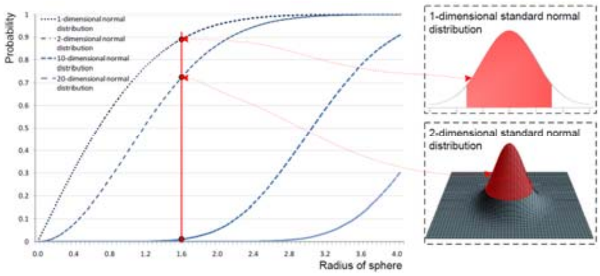
→ in contrast to the low dimensional case, regions of relatively very low density can be extremely important parts

[S86] B.W. Silverman: "Density Estimation for Statistics and Data Analysis". Chapman and Hall/CRC, 1986.

DATABASE SYSTEMS GROUP

Importance of the Tails: Example

- Normal distribution
  (μ= 0, σ= 1)



- 1-dimensional : 90% of the mass of the distribution lies between -1.6 and 1.6
- 10-dimensional: 99% of the mass of the distribution is at points whose distance from the origin is greater than 1.6

→ it is difficult to estimate the density, except for enormous samples

→ in very high dimensions virtually the entire sample will be in the tails

DATABASE SYSTEMS GROUP

Required Sample Sizes for Given Accuracy

- Consider f a multivariate normal distribution
- The aim is to estimate f at the point 0
- The relative mean square error should be fairly small:

$$\frac{E\left[\hat{f}(0) - f(0)\right]^2}{f(0)^2} < 0.1$$

Dimensionality	Required sample size
1	4
2	19
5	768
8	43700
10	842000

→ in the 1,2-dimensional space the given accuracy is obtained from very small samples, whereas in the 10-dimensional space nearly a million observations are required

[S86] B.W. Silverman: "Density Estimation for Statistics and Data Analysis". Chapman and Hall/CRC, 1986.

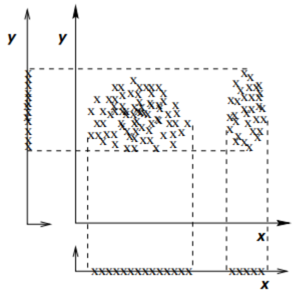
Outline

Feature selection

- A task to remove irrelevant and/or redundant features
  - *Irrelevant* features: not useful for a given task
    - Relevant vs irrelevant
  - *Redundant* features: a relevant feature may be redundant in the presence of another relevant feature with which it is strongly correlated.
- Deleting irrelevant and redundant features can improve the *efficiency* as well as the *quality* of the found methods and patterns.
- New feature space: Delete all useless features from the original feature space.
- Feature selection ≠ Dimensionality reduction
- Feature selection ≠ Feature extraction

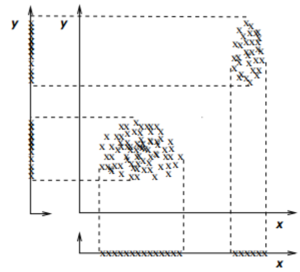
Irrelevant and redundant features (unsupervised learning case)

- Irrelevance



Feature *y* is irrelevant, because if we omit *x*, we have only one cluster, which is uninteresting.

- Redundancy

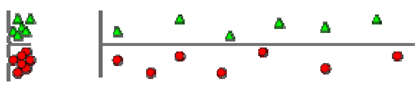


Features *x* and *y* are redundant, because *x* provides the same information as feature *y* with regard to discriminating the two clusters

Source: Feature Selection for Unsupervised Learning, Dy and Brodley, Journal of Machine Learning Research 5 (2004)

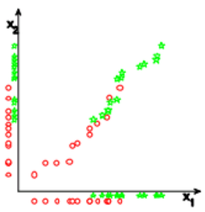
Irrelevant and redundant features (supervised learning case)

- Irrelevance



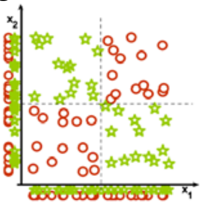
Feature *y* separates well the two classes.
Feature x is irrelevant.
Its addition “destroys” the class separation.

- Redundancy



Features *x* and *y* are redundant.

- Individually irrelevant, together relevant



Source: http://www.kdnuggets.com/2014/03/machine-learning-7-pictures.html

Problem definition

- **Input:** Vector space $F = d_1 \times .. \times d_n$ *with* dimensions $D = \{d_1, .., d_n\}$.
- **Output:** a *minimal* subspace M over dimensions D` $\subseteq$ D which is *optimal* for a giving data mining task.
  - Minimality increases the efficiency, reduces the effects of the curse of dimensionality and increases interpretability.

Challenges:

- Optimality depends on the given task
- There are 2^d possible solution spaces (exponential search space)
- There is often no monotonicity in the quality of subspace (Features might only be useful in combination with certain other features)

$\Rightarrow$ For many popular criteria, feature selection is an exponential problem
$\Rightarrow$ Most algorithms employ search heuristics

2 main components

1. Feature subset generation
   - Single dimensions
   - Combinations of dimensions (subpaces)

2. Feature subset evaluation
   - Importance scores like information gain, χ^2
   - Performance of a learning algorithm

Feature selection methods 1/4

- Filter methods
  - Explores the general characteristics of the data, independent of the learning algorithm.
- Wrapper methods
  - The learning algorithm is used for the evaluation of the subspace
- Embedded methods
  - The feature selection is part of the learning algorithm

Feature selection methods 2/4

- Filter methods
  - Basic idea: assign an ``importance" score to each feature to filter out the useless ones
  - Examples: information gain, χ^2-statistic, TF-IDF for text
  - Disconnected from the learning algorithm.
  - Pros:
    - Fast
    - Simple to apply
  - Cons:
    - Doesn't take into account interactions between features
    - Individually irrelevant features, might be relevant together (recall slide 14)

Feature selection methods 3/4

- Wrapper methods
  - A learning algorithm is employed and its performance is used to determine the quality of selected features.
  - Pros:
    - the ability to take into account feature dependencies.
    - interaction between feature subset search and model selection
  - Cons:
    - higher risk of overfitting than filter techniques
    - very computationally intensive, especially if building the classifier has a high computational cost.

Feature selection methods 4/4

- Embedded methods
  - Such methods integrate the feature selection in model building
  - Example: decision tree induction algorithm: at each decision node, a feature has to be selected.
  - Pros:
    - less computationally intensive than wrapper methods.
  - Cons:
    - specific to a learning method