Freie Universität Berlin

Lecture: Machine Learning for Data Science

Winter semester 2021/22

Lecture 20: High dimensionality (feature selection)

Prof. Dr. Eirini Ntoutsi

What you will learn in this course?

See lecture 1

- Introduction
- Part 1: Basic ML tasks
  - ❑ Supervised learning
  - ❑ Unsupervised learning
  - ❑ Reinforcement learning
  - ❑ Outlier detection
- Part 2: ML for particular/modern data challenges
  - ❑ High-dimensional learning
  - ❑ Learning over non-stationary data
  - ❑ Label/Data scarsity

Part 2: Mind the data characteristics

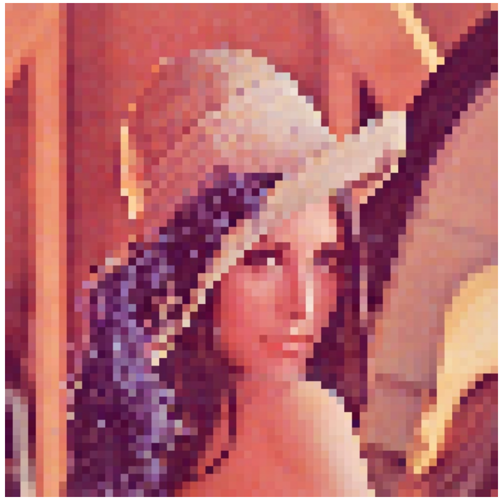
See lecture 1

- In the first part, several assumptions are made about the data
  - Data stationarity (the data distribution does not change)
  - i.i.d. distribution (all instances are independent and identically distributed)
  - Availability of labels (for supervised learning)
  - Low-dimensionality
  - …
- In the second part, we will waive some of these assumptions taking into account (real/modern) data challenges), namely
  - High-dimensionality
  - Non-stationarity
  - Label/Data scarcity
- We will discuss how ML methods (part 1) are affected by these challenges and new methods and algorithms for the particular data challenges

Outline

- Introduction to high dimensional data and challenges of high dimensionality
- How we deal with high dimensionality
- The feature selection task
- Building components of feature selection methods
- Forward Selection and Feature Ranking
- Backward Elimination and Random Subspace Selection
- *k*-dimensional subspace projections
- Overview and discussion
- Things you should know from this lecture & reading material

Examples of high-dimensional data 1/2



Source: https://en.wikipedia.org/wiki/Lenna

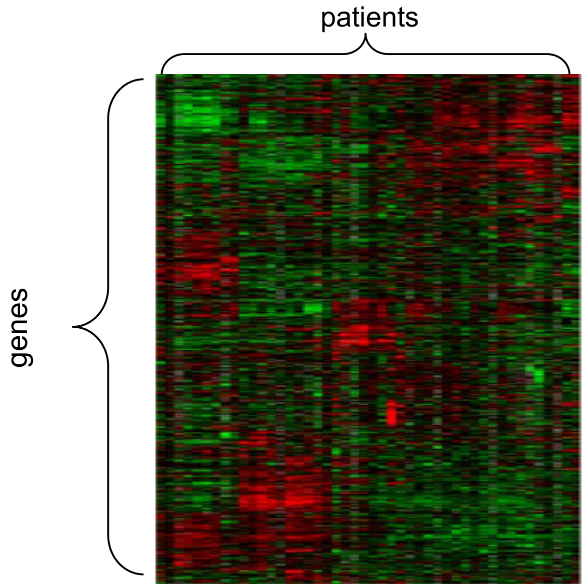
- Image data
  - If *each pixel is a feature*, then a 64x64 image → 4,096 features
  - low-level image descriptors (e.g., color histograms)
  - Typically, regional descriptors

- Metabolome data
  - Metabolomics is the scientific study of chemical processes involving metabolites
  - The term metabolite usually is restricted to small molecules, that are intermediates and products of metabolism[1].
  - *feature = concentration of a metabolite*
  - The Human Metabolome Database contains 41,993 metabolite entries
  - Example: Bavaria newborn screening (For each newborn in Bavaria, the blood concentrations of 43 metabolites are measured in the first 48 hours after birth)

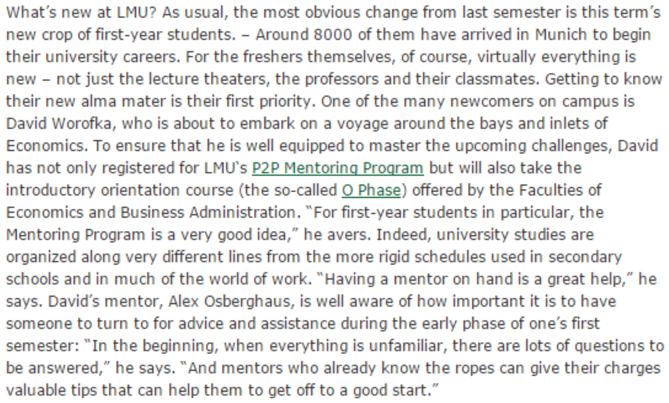
[1] :https://en.wikipedia.org/wiki/Metabolite

Examples of High-Dimensional Data 2/2

- Gene expression data
  - Thousands or tens of thousands of genes in a single experiment
  - *Features correspond to genes*
  - Dimensionality is much higher than the sample size



- Text data
  - *Features correspond to words (unigrams, bigrams ….)*
  - Different documents have different words
  - Very often, esp. in social media, new words are created, like
    - Hashtags
    - Abbreviations (e.g., Dr for doctor)
    - colloquial language (e.g., luv for love)
    - Special words (e.g, hashtags like #uselections, @TwitterUser like @UniHannover)



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: http://tinyurl.com/qhq6byz

Challenges due to high dimensionality 1/8

> *“After overfitting, the biggest problem in machine learning is the curse of dimensionality”*, P. Domingos, Communications of the ACM, 2012.

- Curse of dimensionality: refers to the problem of finding structure in data embedded in a highly dimensional space.
- Original term was coined by Bellman in 1961 to refer to the fact that many algorithms become intractable in high dimensional spaces.
  - A problem is called intractable iff there is no efficient algorithm that solves it.
- For machine learning it refers to much more.

Challenges due to high dimensionality 2/8

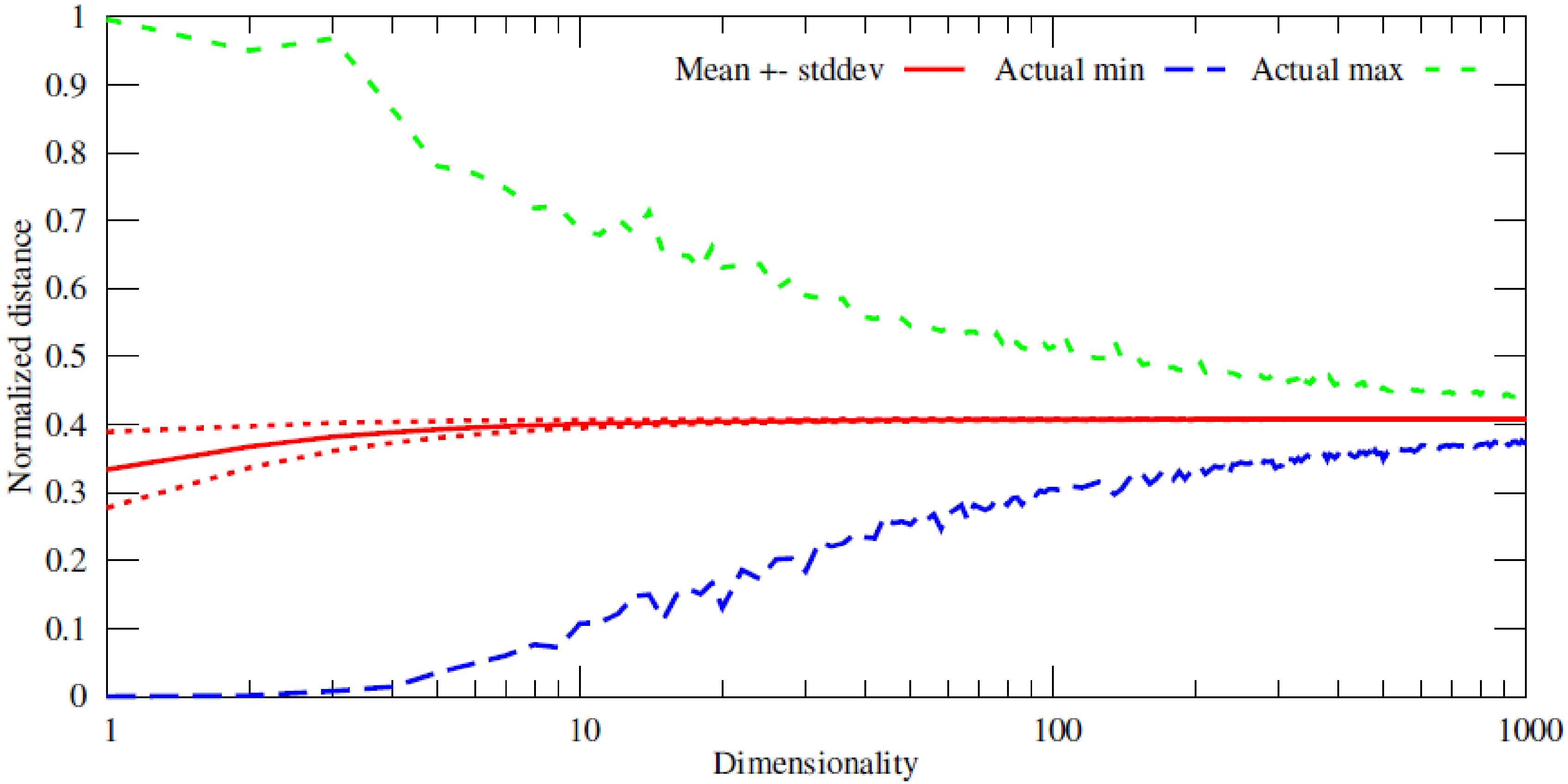
- Challenge 1: The *distance/similarity functions* loose their discriminative power
  - i.e., distance to the nearest and to the farthest neighbor converge

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

- As a consequence, points become equidistant and the choice of the nearest neighbors becomes random.

Challenges due to high dimensionality 3/8

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



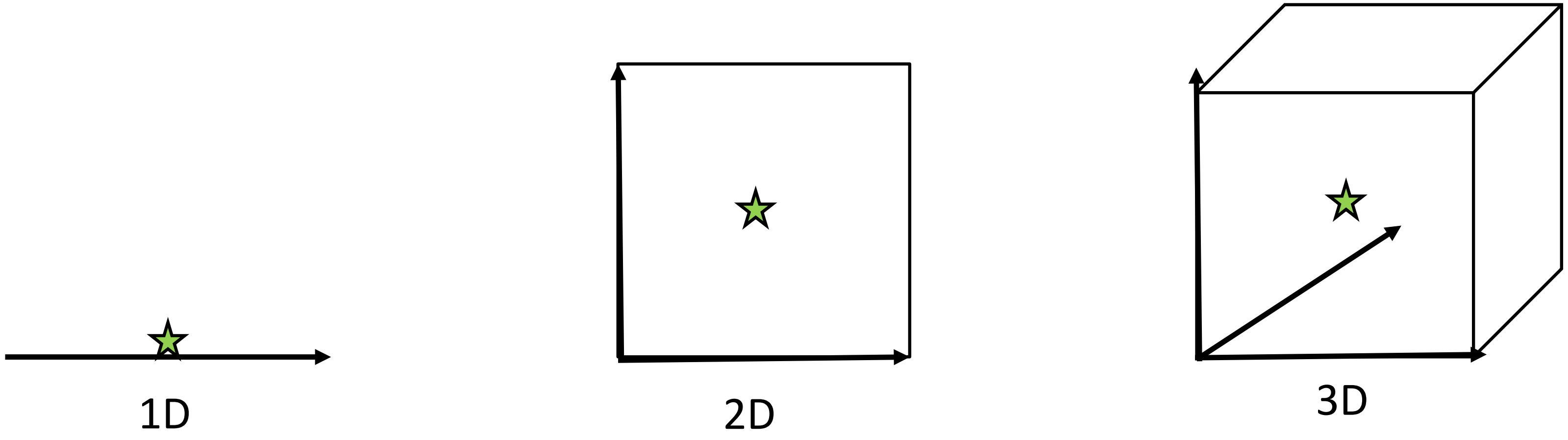
Source: Tutorial on Outlier Detection in High-Dimensional Data, Zimek et al, ICDM 2012

Challenges due to high dimensionality 4/8

- Challenge 2: The *noise* from the irrelevant features completely dominates the *signal* from the relevant ones
  - Consider the feature space of *d relevant* features for a given application (e.g., d=<x_1,x_2>)
  - Now add $d*x$ irrelevant features (e.g., d=<x_1,x_2,x_3,….,x_{100})
  - With increasing *x,* the irrelevant features will determine the final distance/similarity

Challenges due to high dimensionality 5/8

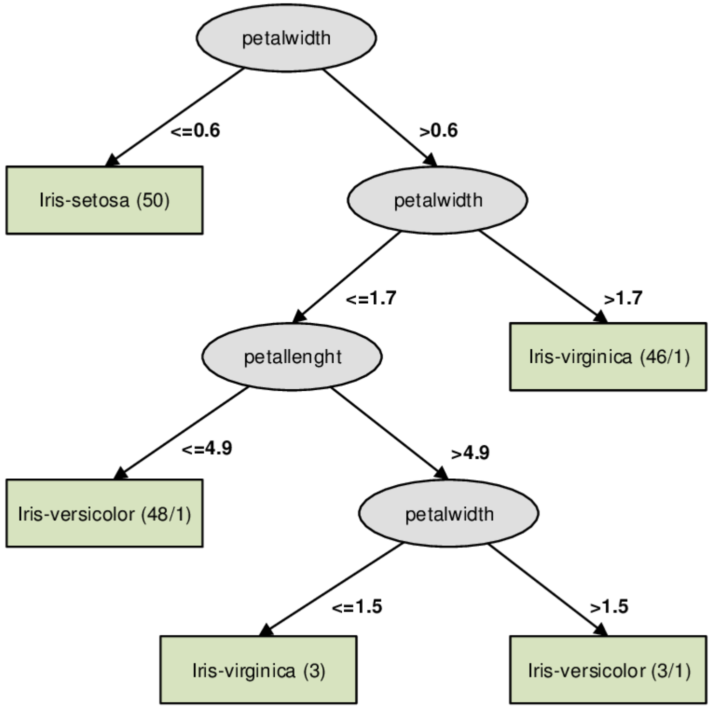
- Challenge 3: The more features, the larger the *hypothesis space* $H = \{ f: X \rightarrow Y \}$.



- Recall that learning is a search/optimization problem over the hypothesis space *H*
- The larger the hypothesis space
  - the harder to find the correct hypothesis
  - the more training instances we need to generalize correctly

Challenges due to high dimensionality 6/8

- Challenge 4: Patterns and models on high-dimensional data are often *hard to interpret*.
  - e.g., long decision tree paths
  - cliques of correlated features dominate the object description



Challenges due to high dimensionality 7/8

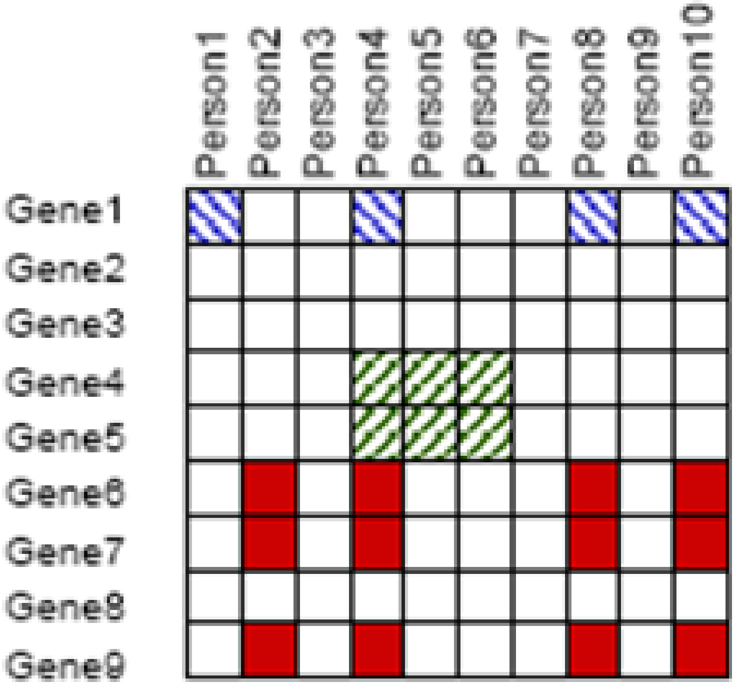
- Challenge 5: *Efficiency* in high-dimensional spaces is often limited
  - index structures degenerate
  - distance computations are much more expensive
    - E.g., Euclidean distance between two data points

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

Challenges due to high dimensionality 8/8

- Challenge 6: Pattern might only be observable in *subspaces* (or *projected spaces)* and not in the full dimensional space



	Person1	Person2	Person3	Person4	Person5	Person6	Person7	Person8	Person9	Person10
Gene1	▧			▧				▧		▧
Gene2										
Gene3										
Gene4				▨	▨	▨				
Gene5				▨	▨	▨				
Gene6		🟥		🟥				🟥		🟥
Gene7		🟥		🟥				🟥		🟥
Gene8										
Gene9		🟥		🟥				🟥		🟥

Outline

- Introduction to high dimensional data and challenges of high dimensionality
- How we deal with high dimensionality
- The feature selection task
- Building components of feature selection methods
- Forward Selection and Feature Ranking
- Backward Elimination and Random Subspace Selection
- *k*-dimensional subspace projections
- Overview and discussion
- Things you should know from this lecture & reading material

Dealing with high dimensionality

- Let $F=\{f_1,f_2,\ldots,f_d\}$ be the high dimensional feature space
- Different approaches for dealing with high dimensionality
  - Feature selection approaches: Find a subset $F' \subset F$ of features that are the most relevant for learning.
    - This lecture
  - Dimensionality reduction approaches: Find a lower dimensional data representation F'' that still preserves properties of the data. F'' consists of "combinations" of the original features
    - Next lecture
  - Learning in subspaces: look for data in subspaces, instead of the full feature space
    - Will not be covered
  - …

Outline

- Introduction to high dimensional data and challenges of high dimensionality
- How we deal with high dimensionality
- The feature selection task
- Building components of feature selection methods
- Forward Selection and Feature Ranking
- Backward Elimination and Random Subspace Selection
- *k*-dimensional subspace projections
- Overview and discussion
- Things you should know from this lecture & reading material

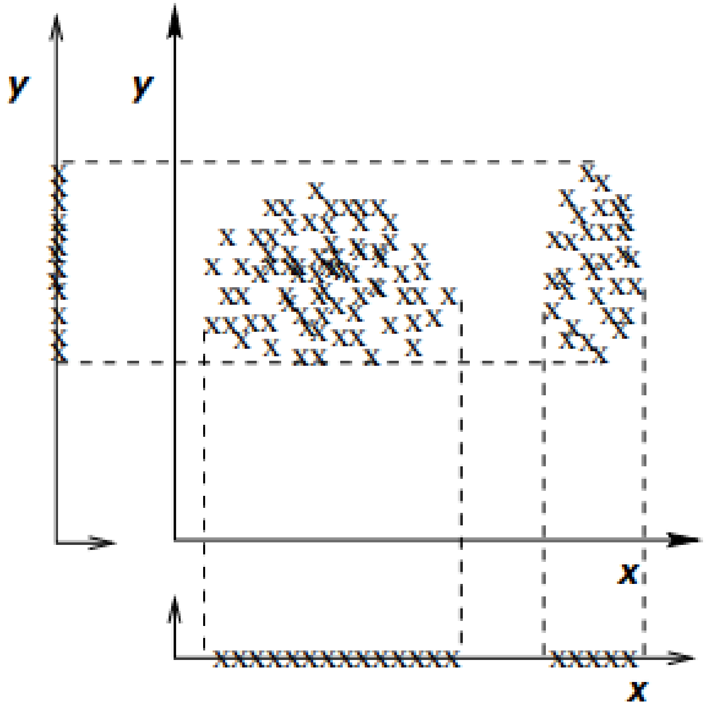
The feature selection task

- Given a problem in the original feature space *F* and a learning task (supervised or unsupervised)
- Goal of the feature selection task: remove irrelevant and/or redundant features from *F*
  - *Irrelevant* features: not useful for the learning task
  - *Redundant* features: a relevant feature may be redundant in the presence of another relevant feature with which it is strongly correlated.
- The result is a new feature space $F' \subseteq F$, where all “useless” features from *F* have been removed.
- Removing irrelevant and redundant features can improve the efficiency, the quality of the models, the interpretability of the results and in general, reduce the effects of high dimensionality.

- Attention:
  - Feature selection ≠ Feature extraction
  - Feature selection ≠ Dimensionality reduction

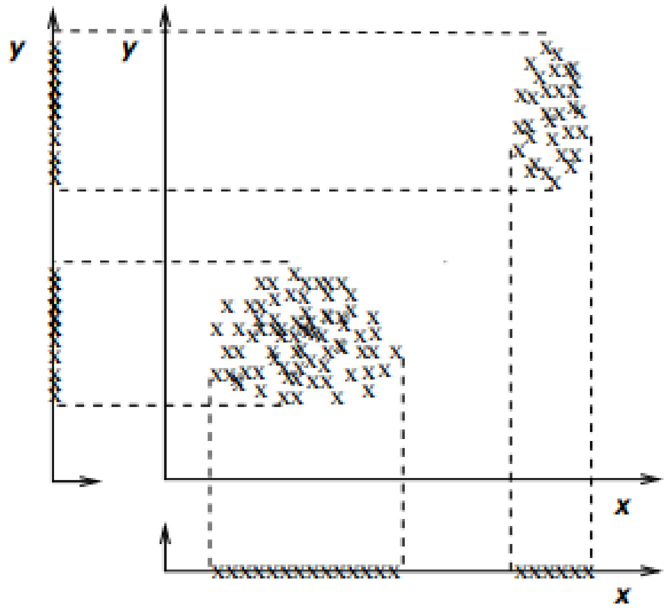
Irrelevant and redundant features: unsupervised learning case

- Irrelevance



Feature *x* is relevant, it separates well the two groups. Feature *y* is irrelevant, because if we omit *y* we get the same clustering (due to *x*).

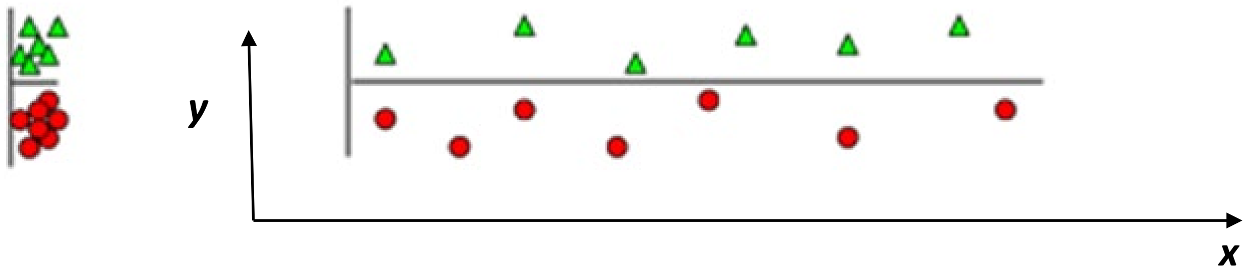
- Redundancy



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

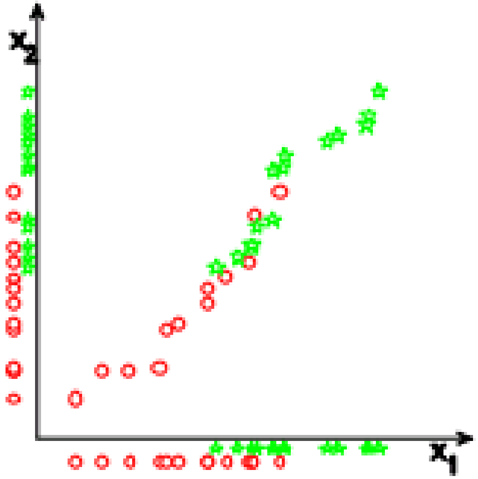
Irrelevant and redundant features: supervised learning case 1/2

- Irrelevance



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

- Redundancy

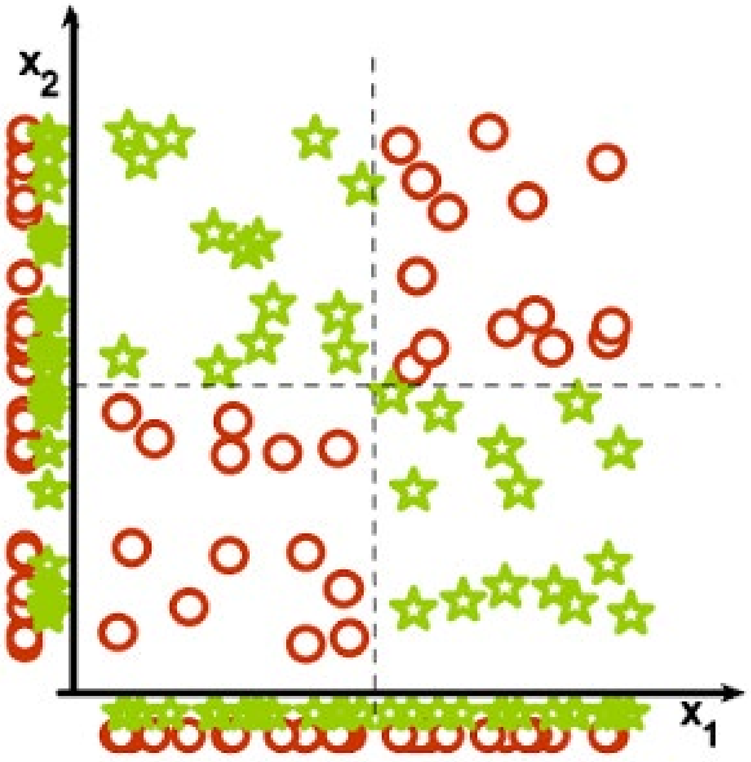


Features x_1 and x_2 are redundant.

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

Irrelevant and redundant features: supervised learning case 2/2

- Even worse, individually irrelevant features, might be relevant together!!!

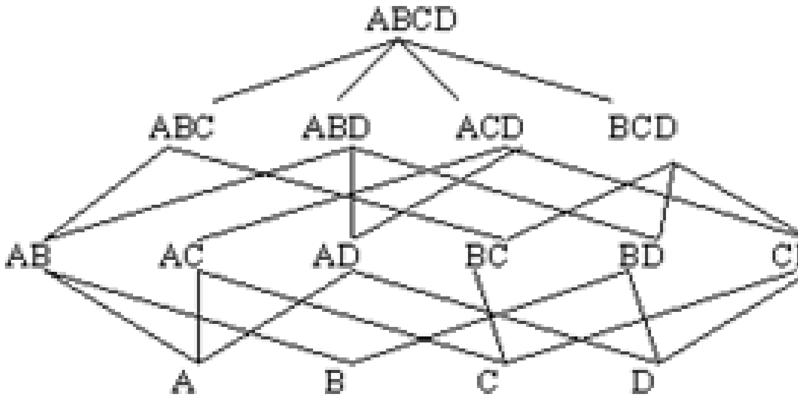


Formal problem definition

- **Input:** A dataset *D* described in a (high) *d*-dimensional feature space $F = \{f_{1,}..,f_d\}$.
- **Output:** a minimal subspace dimensions $F` \subseteq F$ which is optimal for a giving learning task.

Challenges:

- *Optimality* depends on the given task (supervised, unsupervised)
- *Minimality*: 2^d possible solution spaces (*exponential* search space)
- There is often no monotonicity in the quality of a subspace
  - No easy prunning



⇒ For many popular criteria, feature selection is an exponential problem

⇒ Most algorithms employ search heuristics

⇒ In many cases, the desired cardinality $k=|F'|$ is given by the user

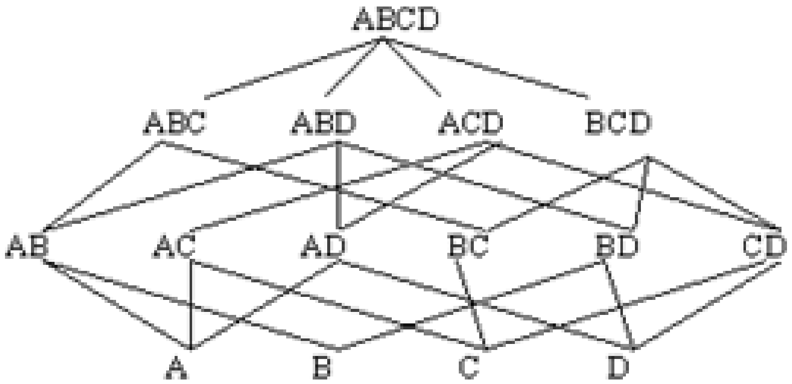
Outline

- Introduction to high dimensional data and challenges of high dimensionality
- How we deal with high dimensionality
- The feature selection task
- Building components of feature selection methods
- Forward Selection and Feature Ranking
- Backward Elimination and Random Subspace Selection
- *k*-dimensional subspace projections
- Overview and discussion
- Things you should know from this lecture & reading material

2 main components in feature selection methods

1. Feature subset generation (or, which subspaces to check?)
   - Single dimensions, e.g., *A*, *B*
   - Combinations of dimensions (subspaces), e.g., *AB*, *ACD*

2. Feature subset evaluation (or, how to evaluate a subspace?)
   - Importance scores like information gain, χ^2
   - Performance of a learning algorithm on the subspace



3 categories of feature selection methods

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

1. Filter methods

- Explore the general characteristics of the data, independent of the learning algorithm.
    - 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 29)

2. Wrapper methods

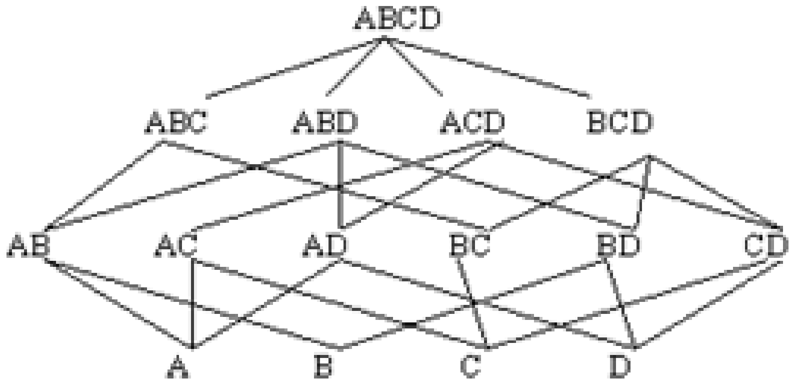
- The learning algorithm is used for the evaluation of the subspace
  - Basic idea: A learning algorithm is employed and its performance is used to determine the quality of selected features.
- Pros:
  - the ability to take into account feature dependencies.
  - interaction between feature subset search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

3. Embedded methods

- Feature selection is part of the learning algorithm
  - Basic idea: Integrate feature selection into the model building process
  - Example: decision tree induction algorithm: at each decision node, a feature is selected as the splitting attribute.
- Pros:
  - less computationally intensive than wrapper methods.
- Cons:
  - specific to a learning method

Different search strategies in the feature space

- Forward selection
  - Start with an empty feature space and add relevant features
- Backward selection
  - Start with all features and remove irrelevant features
- Branch-and-bound backward selection search
  - Find the optimal subspace under the monotonicity assumption
- Randomized search
  - Randomized search for a *k* dimensional subspace
- ...



Outline

- Introduction to high dimensional data and challenges of high dimensionality
- How we deal with high dimensionality
- The feature selection task
- Building components of feature selection methods
- Forward Selection and Feature Ranking
- Backward Elimination and Random Subspace Selection
- *k*-dimensional subspace projections
- Overview and discussion
- Things you should know from this lecture & reading material

1. Forward Selection and Feature Ranking

ID	Age	Car type	Risk
1	23	Familie	high
2	17	Sport	high
3	43	Sport	high
4	68	Familie	low
5	32	LKW	low

Predictor attributes

Class attribute

- **Input**: A supervised learning task
  - A training set *D*={(*X*, *y*)} from the *FxY* space
    - Attributes space: $F=\{f_1,f_2,\ldots,f_d\}$
    - Class attribute: $Y=\{y_1,\ldots,y_k\}$
    - $\forall X \in \mathrm{D}: \mathrm{X} = < x_1, x_2, \ldots, x_d >$, $y \in \mathrm{Y}$
- **Approach**
  - Compute how ``*relevant*” each dimension $f_i \in F$ is in predicting the class attribute *Y*, i.e., $quality(f_{i\,,}Y)$
  - Sort the dimensions $f_{1,},..,f_d$ w.r.t. their quality $quality(f_{i\,,}Y)$
  - Select the *k*-best dimensions
- Assumption:
  - Features are only correlated via their connection to *Y*

=> therefore, it is sufficient to evaluate the connection between each single feature f_i and the target variable *Y*

Quality measures for features

- The quality measure *quality($f_{i,}$Y)* evaluates how suitable is a feature f_i for predicting the value of the class attribute *Y*.
- Example quality measures:
  - Information Gain
  - Chi-square χ^2-statistics
  - Mutual Information
  - …

Most of these measures were covered in the first part: We will quickly mention them here, but we are not going to repeat them . Please check previous lecture slides if needed.

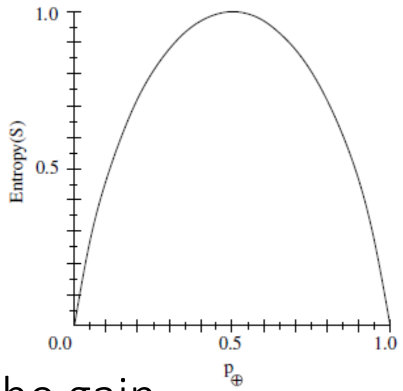
Information Gain 1/2

See lecture 3

- Idea: Evaluate class discrimination in each dimension (Used in ID3 algorithm)
- It uses entropy, a measure of pureness of the data

$$Entropy(S) = \sum_{i=1}^{k} - p_i \log_2(p_i)$$

(p_i : relative frequency of *class* c_i in *S*)



- The information gain Gain(S,A) of an attribute A relative to a training set S measures the gain reduction in S due to splitting on A:

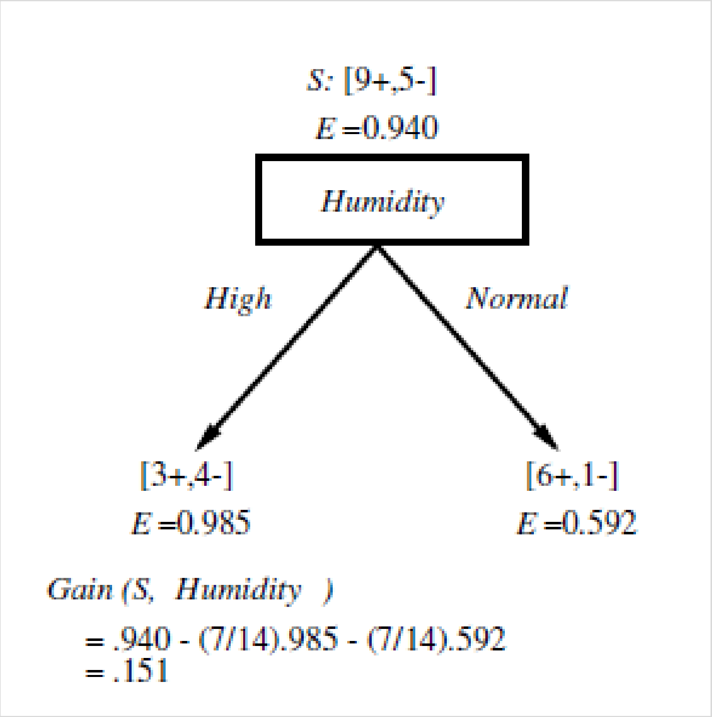
$$Gain(S,A) = \underbrace{\boxed{Entropy(S)}}_{\text{Before splitting}} - \underbrace{\boxed{\sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)}}_{\text{After splitting on A}}$$

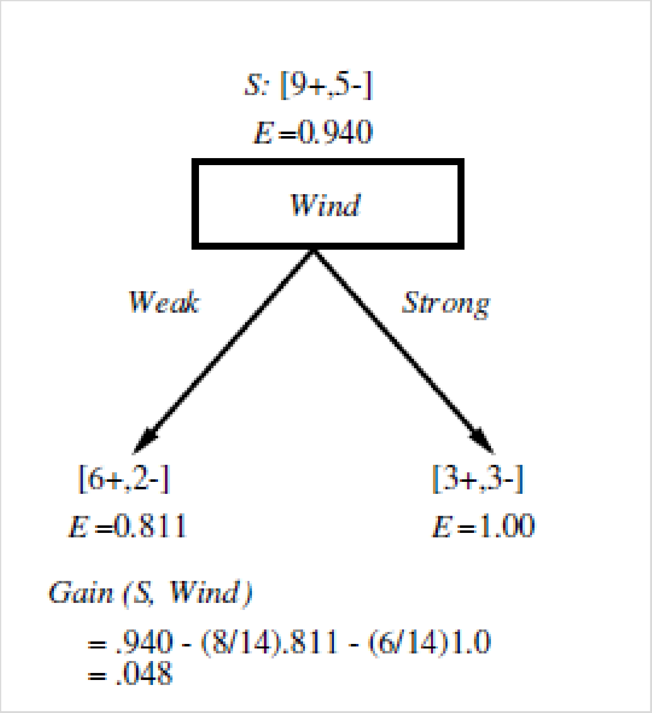
- For nominal attributes: use attribute values
- For real valued attributes: Determine a splitting position in the value set.
- Larger gain values better!

Information Gain 2/2

See lecture 3

- Which attribute, “Humidity” or “Wind” is better?





Chi-square χ2 statistics 1/2

See lecture 3

- Idea: Measures the independency of a variable from the class variable.
- Contingency table: Divide data based on a split value s or based on discrete values
- Example: Liking science fiction movies implies playing chess?

Class attribute

Predictor attribute

	Play chess	**Not play chess**	Sum (row)
Like science fiction	250	200	450
Not like science fiction	50	1000	1050
Sum(col.)	300	1200	1500

- Chi-square χ^2 test

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

o_{ij}:observed frequency
e_{ij}: expected frequency

$$e_{ij} = \frac{h_i h_j}{n}$$

- Larger values better!

Forward Selection and Feature Ranking - overview

- Advantages:
  - Efficiency: it evaluated $\{f_1, f_2, \ldots, f_d\}$ features w.r.t. the class attribute Y instead of $\binom{d}{k}$ subspaces

- Disadvantages:
  - Independency assumption: Features must display direct correlations to the classes
  - In case of correlated features: Always selects the features having the strongest direct correlation to the class variable, even if the features are strongly correlated with each other.
    - So the resulting subspace, might contain highly correlated or even redundant features.

Outline

- Introduction to high dimensional data and challenges of high dimensionality
- How we deal with high dimensionality
- The feature selection task
- Building components of feature selection methods
- Forward Selection and Feature Ranking
- Backward Elimination and Random Subspace Selection
- *k*-dimensional subspace projections
- Overview and discussion
- Things you should know from this lecture & reading material

2. Backward Elimination

- **Idea**: Start with the complete feature space and delete redundant features

```
Algorithm: Greedy Backward Elimination
Generate the subspaces R of the feature space F
Evaluate subspaces R with the quality measure quality(R)
Select the best quality subspace, R*
If R* has the wanted dimensionality, terminate
     else start backward elimination on R*.
```

- Useful in supervised and unsupervised setting
- Subspace quality evaluation: in unsupervised cases, *quality*(*R*) measures structural characteristics
- Subspace generation:
  - Greedy search if there is no monotonicity on *quality*(*R*); for monotonous *quality*(*R*), branch and bound search

Subspace quality measures: Distance-based subspace quality

- **Idea:** The subspace quality can be evaluated by the distance between the within-class nearest neighbor and the between-classes nearest neighbor
- For each $X \in D$ *with class label* $Y=y$, compute the closest object having the same class $NN^U_y(X)$ (*within-class nearest neighbor*) and the closest object belonging to another class $NN^U_{y' \neq y}(X)$ (*between-classes nearest neighbor*) in the subspace U under investigation.
- The distance-based subspace quality is defined as:

$$q(U) = \frac{1}{|D|} \cdot \sum_{X \in D} \frac{NN^U_{y' \neq y}(X)}{NN^U_y(X)}$$

- Remark: $q(U)$ is not monotonous.
  - By deleting a dimension, the quality can increase or decrease.

Subspace quality measures: Model-based approach

- **Idea**: Directly employ a data mining algorithm to evaluate the subspace.
- An example: Evaluate each subspace using a Naive Bayes classifier

- Practical aspects:
  - The evaluation of the subspace via the learning algorithm is very important
    - Test set should be representative of the problem
    - Success of the learning algorithm must be measurable (e.g. accuracy)
  - Runtime for training and applying the classifier should be low
  - The classifier parameterization should not be of great importance

Backward Elimination – with greedy subspace generation

Advantages:

- Considers complete subspaces (multiple dependencies are used)
- Can recognize and eliminate redundant features

Disadvantages:

- Tests w.r.t. subspace quality usually requires much more effort
- All solutions employ heuristic greedy search which do not necessarily find the optimal feature space.

Backward elimination: Branch and Bound Search

- The backward elimination approach "Branch and Bound", by Narendra and Fukunaga, 1977 is guaranteed to find the optimal *k*-dimensional feature subset under the monotonicity assumption
- The monotonicity assumption states that for two subsets *X*, *Y* and a feature selection criterion function *J*, if:

$$X \subset Y \Rightarrow \mathrm{J(X)} < \mathrm{J(Y)}$$

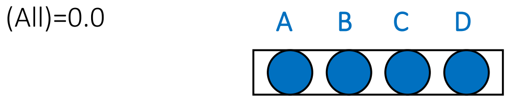
  - E.g. $X=\{d_1,d_2\}$, $Y=\{d_1,d_2,d_3\}$

- Branch and Bound starts from the full set and removes features using *a depth-first strategy (DFS)*
  - Nodes whose objective function are lower than the current best are not explored since the monotonicity assumption ensures that their children will not contain a better solution.

Example: Branch and Bound Search 1/8

Example: Original dimensionality 4, <A,B,C,D>. Target dimensionality *d* = 1.

● selected feature ○ removed feature

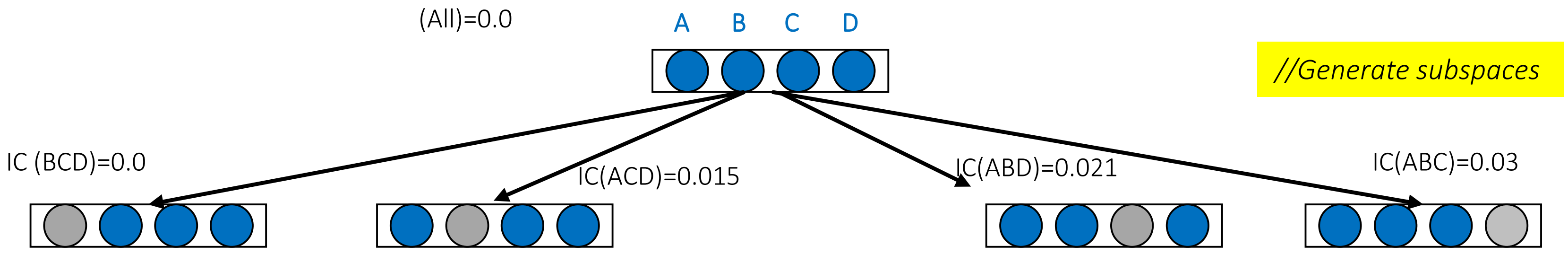


//Start from the full set

Example: Branch and Bound Search 2/8

Example: Original dimensionality 4, <A,B,C,D>. Target dimensionality *d* = 1.

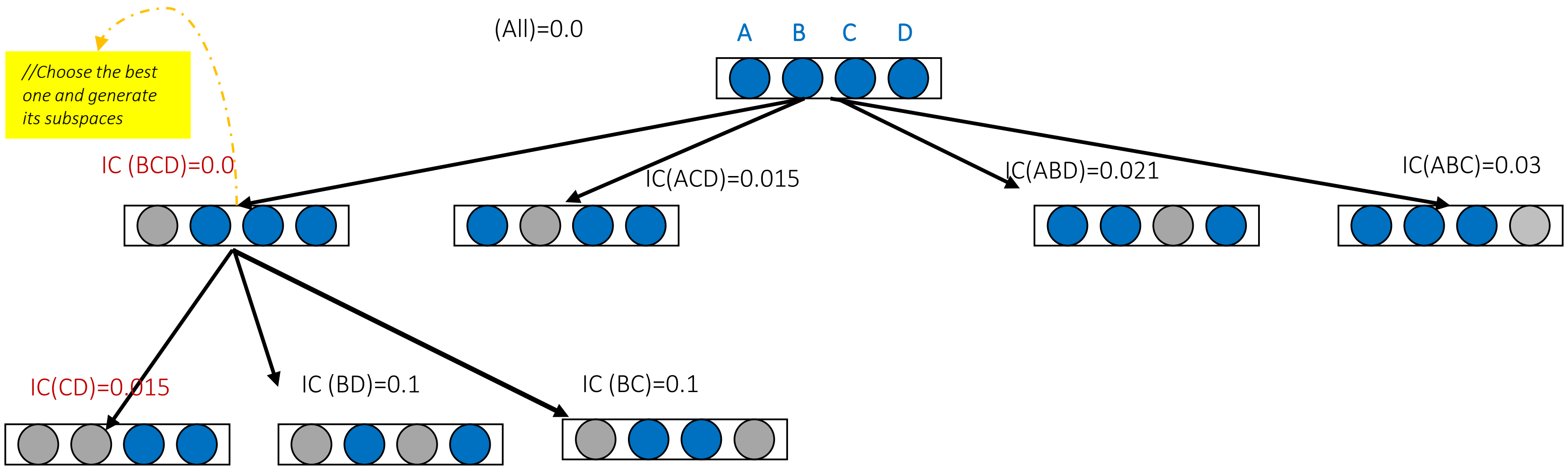
● selected feature ● removed feature



Example: Branch and Bound Search 3/8

Example: Original dimensionality 4, <A,B,C,D>. Target dimensionality *d* = 1.

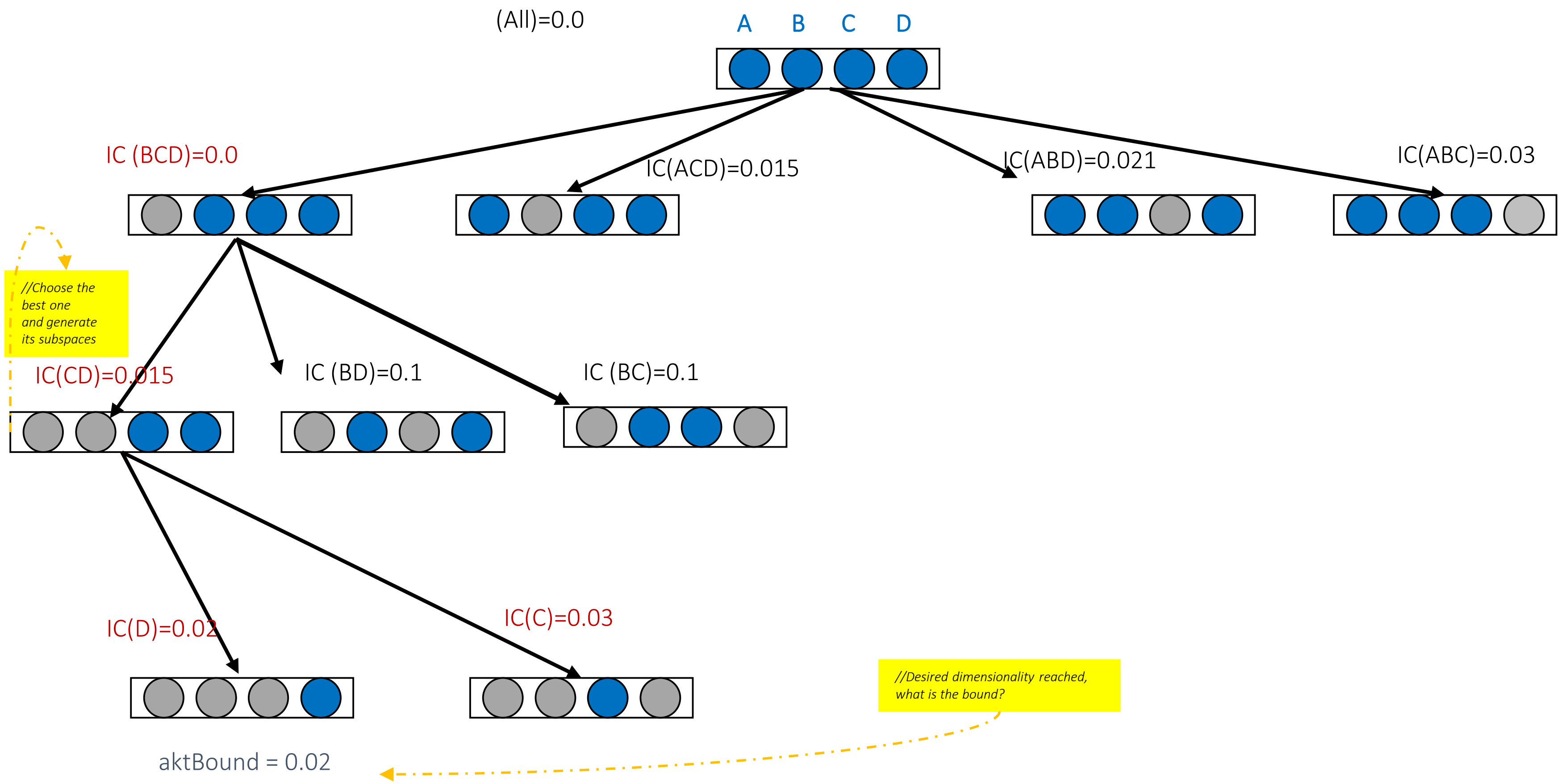
● selected feature ● removed feature



Example: Branch and Bound Search 4/8

Example: Original dimensionality 4, <A,B,C,D>. Target dimensionality *d* = 1.

● selected feature ○ removed feature



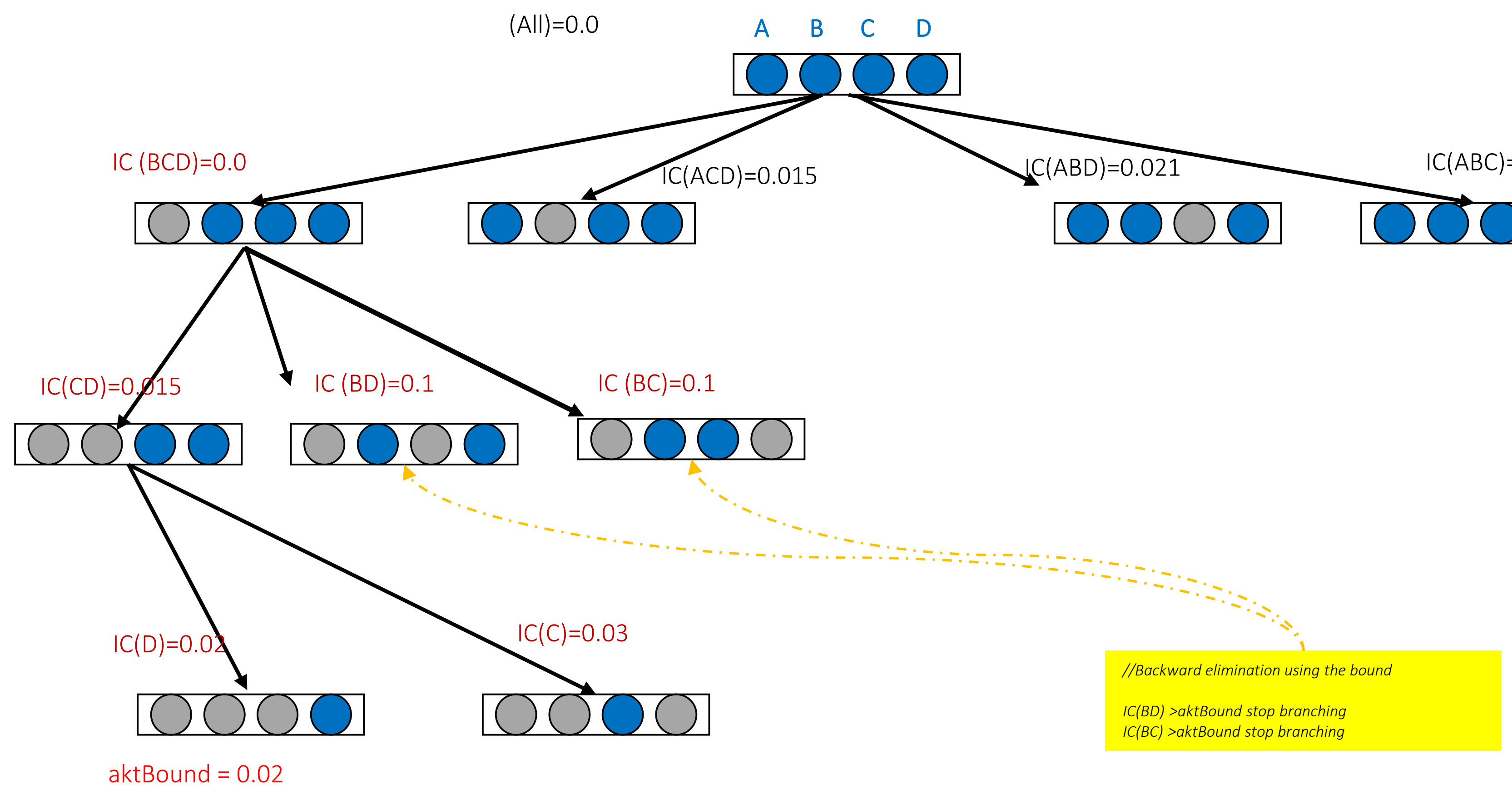
//Choose the best one and generate its subspaces

//Desired dimensionality reached, what is the bound?

Example: Branch and Bound Search 5/8

Example: Original dimensionality 4, <A,B,C,D>. Target dimensionality *d* = 1.

● selected feature ● removed feature



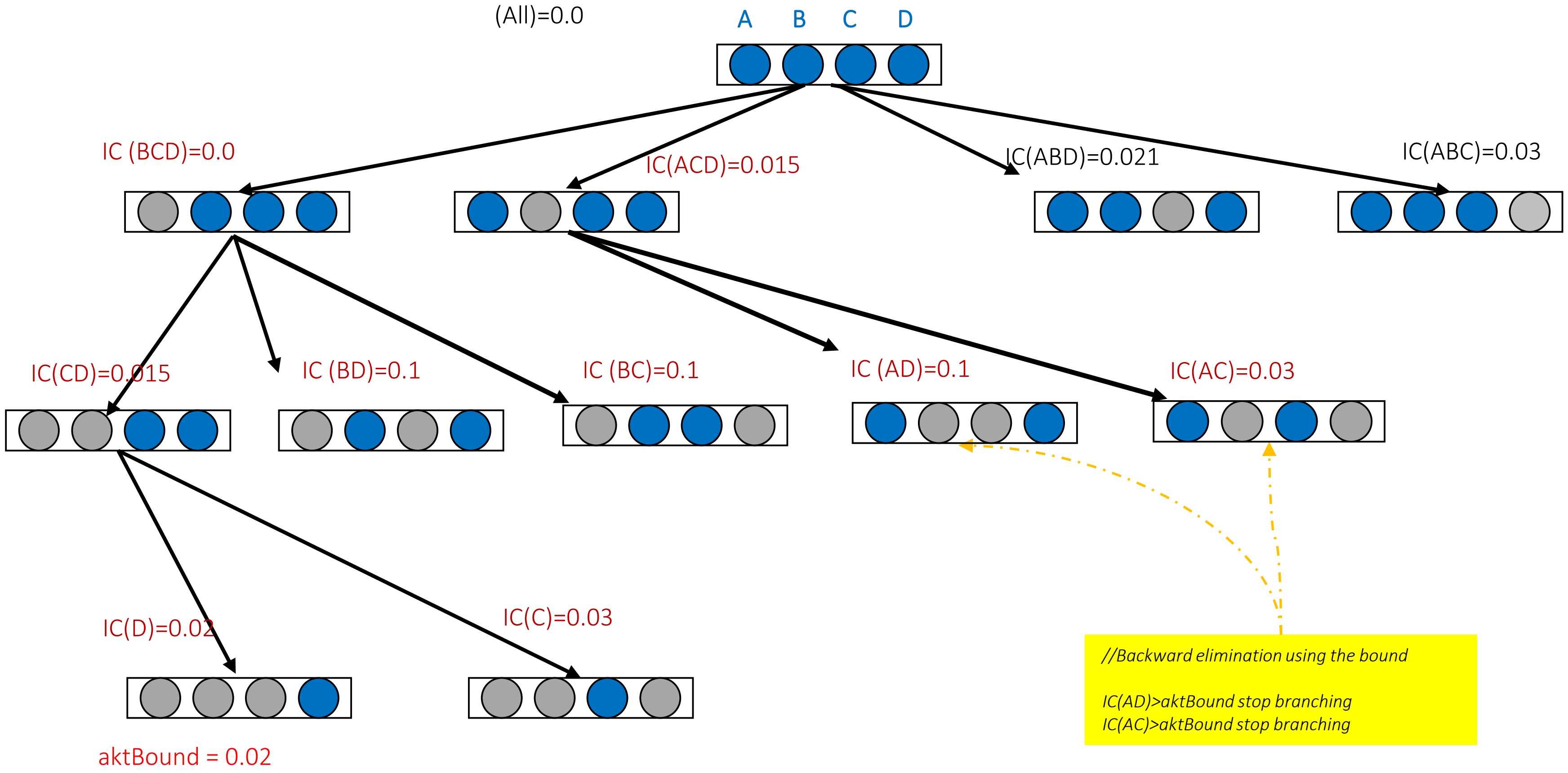
//Backward elimination using the bound

IC(BD) >aktBound stop branching
IC(BC) >aktBound stop branching

Example: Branch and Bound Search 7/8

Example: Original dimensionality 4, <A,B,C,D>. Target dimensionality *d* = 1.

● selected feature ○ removed feature



(All)=0.0

A B C D

IC (BCD)=0.0

IC(ACD)=0.015

IC(ABD)=0.021

IC(ABC)=0.03

IC(CD)=0.015

IC (BD)=0.1

IC (BC)=0.1

IC (AD)=0.1

IC(AC)=0.03

IC(D)=0.02

IC(C)=0.03

aktBound = 0.02

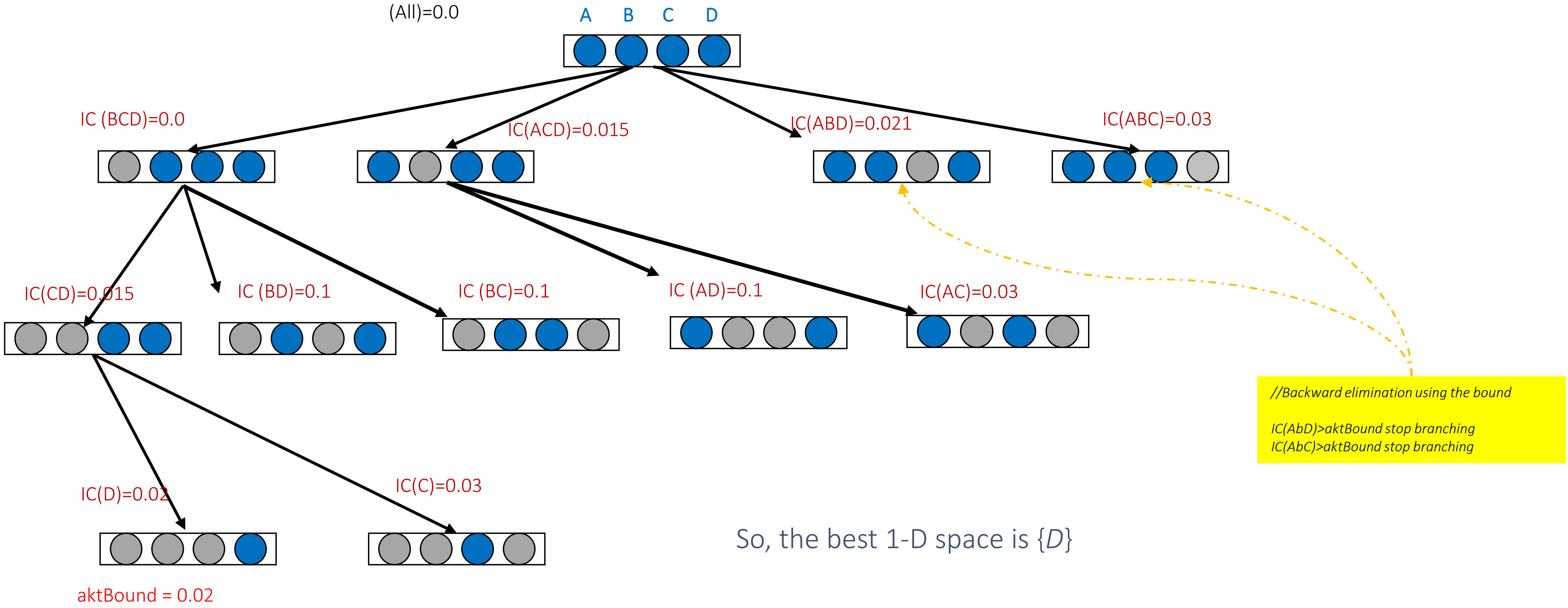
//Backward elimination using the bound

IC(AD)>aktBound stop branching
IC(AC)>aktBound stop branching

Example: Branch and Bound Search 7/8

Example: Original dimensionality 4, <A,B,C,D>. Target dimensionality *d* = 1.

● selected feature ○ removed feature



//Backward elimination using the bound

IC(AbD)>aktBound stop branching
IC(AbC)>aktBound stop branching

So, the best 1-D space is {*D*}

Subspace quality measures: Subspace Inconsistency (IC)

- Let $u, v \in D$. $u=<u_1,u_2,..,u_d>$, $v=<v_1,v_2,..,v_d>$
- **Idea:** Inconsistency means having identical feature vectors *u ,v* ($v_i = u_i \;\; 1 \le i \le d$) in subspace *U* but different class labels ($C(u) \neq C(v)$)
- Measuring the inconsistency of a subspace *U*
  - $X_U(A)$: Number of all identical vectors *A* in *U*
  - $X^c_U(A)$: Number of all identical vectors in *U* having class label *C*
- Inconsistency of *U* w.r.t *A*:

$$IC_U(A) = X_U(A) - \max_{c \in C} X_U^c(A)$$

- Inconsistency of U:

$$IC(U) = \frac{\sum_{A \in F} IC_U(A)}{|F|}$$

- Monotonicity:

$$U_1 \subset U_2 \Rightarrow IC(U_1) \geq IC(U_2)$$

ID	Temperature	Humidity	Class
1	10-20	high	no
2	20-30	low	yes
3	10-20	low	yes
4	20-30	high	no
5	20-30	high	yes

Branch and Bound search - overview

- **Advantage**:
  - Monotonicity allows efficient search for optimal solutions
  - Well-suited for binary or discrete data
    (identical vectors are very likely with decreasing dimensionality)

- **Disadvantages:**
  - Useless without groups of identical features (real-valued vectors)
  - Worse-case runtime complexity remains exponential in *d*

Outline

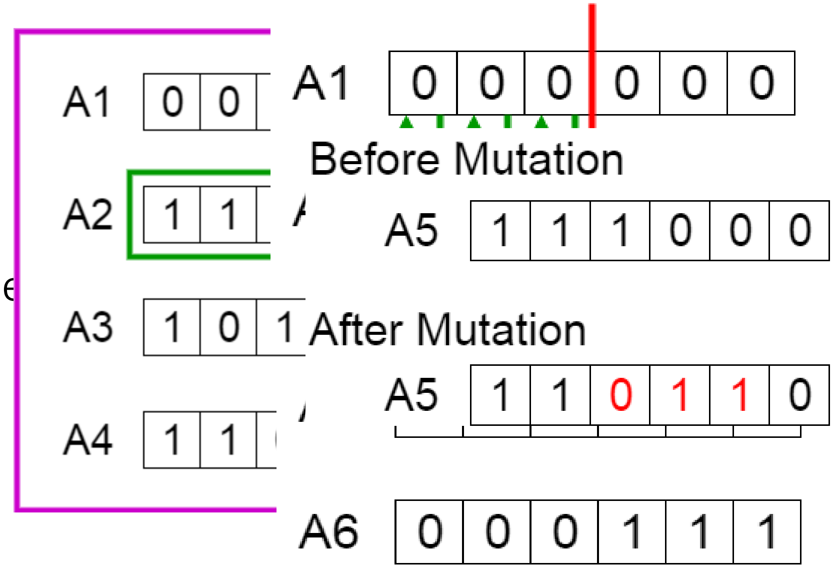
- Introduction to high dimensional data and challenges of high dimensionality
- How we deal with high dimensionality
- The feature selection task
- Building components of feature selection methods
- Forward Selection and Feature Ranking
- Backward Elimination and Random Subspace Selection
- *k*-dimensional subspace projections
- Overview and discussion
- Things you should know from this lecture & reading material

Randomized *k*-dimensional subspace projections

- Idea: Select *n* random subspaces with the target dimensionality *k*=|*F'*|, evaluate each of them and select the best one.
  - Number of possible subspaces: $\binom{d}{k}$
- In practice:
  - Needs quality measures for complete subspaces
  - Trade-off between quality of subspaces and effort depends on parameter *k.*
- Disadvantages:
  - Computational effort and quality of results strongly depend on the subspace quality evaluation measure and the sample size *n*.
  - No directed search for combining highly-relevant and non-redundant features.
- Randomization approaches
  - Genetic algorithms
  - Feature clustering

Genetic Algorithms for feature selection 1/3

- Idea: Randomized search through genetic algorithms
- A genetic algorithm is a search heuristic that is inspired by Darwin's theory of natural evolution
  - It reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation.
- Key building components of a genetic algorithm
  - Encoding of an individual (an individual is a solution to the problem you want to solve): how to represent a solution to the problem
  - Initial population: how to select an initial set of individuals/ solutions
  - Fitness function: evaluates the goodness of an individual/solution
  - Selection: selecting the fittest individuals for the next generation
  - Crossover: exchange the genes of parents among themselves to create
  - Mutation: change slightly existing solutions



Genetic Algorithms for feature selection 2/3

- Genetic algorithms for feature selection
  - Encoding of the individuals: bit-strings
  - Population of solutions: set of k-dimensional subspaces
  - Fitness function: quality measure for a subspace
  - Selection: All subspaces with a quality above some threshold are copied to the next generation.
  - Mutation: dimension d_i in subspace U is replaced by dimension d_j with a likelihood of p%
  - Crossover: combine two subspaces U_1, U_2
    - Unite the features sets of U_1 and U_2.
    - Delete random dimensions until dimensionality is k
  - Free tickets: Additionally each subspace is copied into the next generation with a probability of p'%.

f_1	f_2	f_3	…	f_d
1	0	0	…	1

Feature-clustering 1/2

- Idea: Cluster features in the space of data objects and select one representative feature for each of the clusters
  - This is equivalent to clustering in a transposed data matrix
- Typical example: item-based collaborative filtering

	1 (Titanic)	2 (Braveheart)	3 (Matrix)	4 (Inception)	5 (Hobbit)	6 (300)
Susan	5	2	5	5	4	1
Bill	3	3	2	1	1	1
Jenny	5	4	1	1	1	4
Tim	2	2	4	5	3	3
Thomas	2	1	3	4	1	4

Feature-clustering 2/2

- Algorithmic schema:
  - Cluster features using *k*-medoid clustering based on object similarity
    - A cluster corresponds to a set of dependent features
  - The resulting medoid-features from each cluster comprise the new feature space
- Remarks:
  - Different similarity measures can be used depending on the application
  - Other clustering algorithms could be used as well
- Pros
  - Efficiency (depending on the clustering algorithm)
  - Unsupervised method
- Cons
  - The results depend on the quality of clustering (representatives are usually unstable for different clustering methods and parameters).
  - Results are usually not deterministic (e.g., partitioning clustering)

Outline

- Introduction to high dimensional data and challenges of high dimensionality
- How we deal with high dimensionality
- The feature selection task
- Building components of feature selection methods
- Forward Selection and Feature Ranking
- Backward Elimination and Random Subspace Selection
- *k*-dimensional subspace projections
- Overview and discussion
- Things you should know from this lecture & reading material

Feature selection: overview

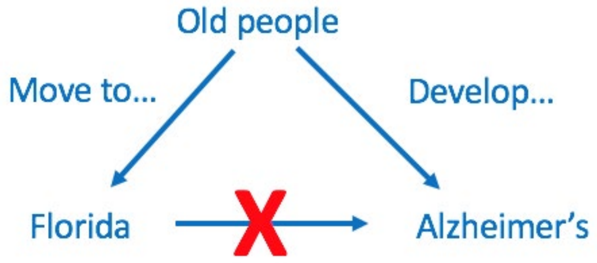
- *Forward-Selection*: Examines each dimension $f_i \in F$ and selects the k-best features
  - Greedy Selection based on Information Gain, $\chi 2$ Statistics or Mutual Information

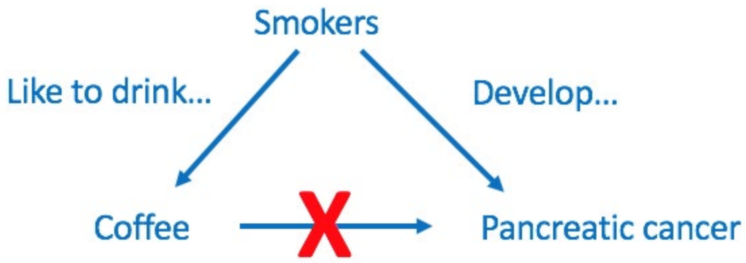
- *Backward-Elimination*: Start with the complete feature space and successively removes the worst dimensions.
  - Greedy elimination with model-based and nearest-neighbor based approaches
  - Optimal elimination via branch and bound search based on inconsistency

- *k-dimensional Projections*: Directly search in the set of k-dimensional subspaces for the best suited subspace
  - Genetic algorithms (fitness function similar to quality measures in backward elimination)
  - Feature clustering based on correlation

Feature selection: discussion

- Many algorithms based on different heuristics
- There are two reason to delete features:
  - Redundancy: Features can be expressed by other features.
  - Missing correlation to the target variable
- Often even approximate results are capable of increasing efficiency ar
- Caution: "*Correlation does not imply causation*"
  - Selected features need not have a causal connection to the target variab depend on the same mechanisms in the data space (hidden variables).
- Different indicators to consider in the comparison of before and after
  - Model performance, time, interpretability/dimensionality, …





Outline

- Introduction to high dimensional data and challenges of high dimensionality
- How we deal with high dimensionality
- The feature selection task
- Building components of feature selection methods
- Forward Selection and Feature Ranking
- Backward Elimination and Random Subspace Selection
- *k*-dimensional subspace projections
- Overview and discussion
- Things you should know from this lecture & reading material

Things you should know from this lecture

- Why feature selection
- Curse of dimensionality
- Forward selection and feature ranking methods
  - Information Gain
  - $\chi 2$ Statistics
- Backward Elimination methods
  - Model-based subspace quality
  - Nearest neighbor-based subspace quality
  - Inconsistency and Branch & Bound search
- k-dimensional projections
  - Genetic algorithms
  - Feature clustering
- Correlation does not imply causation

Hands on experience

- Try different feature selection methods for the Adult dataset

Thank you

Questions/Feedback/Wishes?

Reading material

- I. Guyon, A. Elisseeff: An Introduction to Variable and Feature Selection, Journal of Machine Learning Research 3, 2003.
- P. Domingos, A Few Useful Things to Know about Machine Learning (esp. Section 6), CACM 2012.
- H. Liu and H. Motoda, *Computations methods of feature selection*, Chapman & Hall/ CRC, 2008.
- A.Blum and P. Langley: *Selection of Relevant Features and Examples in Machine Learning*, Artificial Intelligence (97),1997.
- H. Liu and L. Yu: *Feature Selection for Data Mining* (WWW), 2002.
- L.C. Molina, L. Belanche, Â. Nebot: Feature Selection Algorithms: *A Survey and Experimental Evaluations*, ICDM 2002, Maebashi City, Japan.
- P. Mitra, C.A. Murthy and S.K. Pal: *Unsupervised Feature Selection using Feature Similarity*, IEEE Transacitons on pattern analysis and Machicne intelligence, Vol. 24. No. 3, 2004.
- J. Dy, C. Brodley: *Feature Selection for Unsupervised Learning*, Journal of Machine Learning Research 5, 2004.
- M. Dash, H. Liu, H. Motoda*: Consistency Based Feature Selection*, 4th Pacific-Asia Conference, PADKK 2000, Kyoto, Japan, 2000.

Acknowledgements

- The slides are based on
  - *DM2 lecture@LUH(@Eirini Ntoutsi), KDD2/ SS16 lecture@LMU Munich (@Eirini Ntoutsi, Matthias Schubert, Arthur Zimek)*