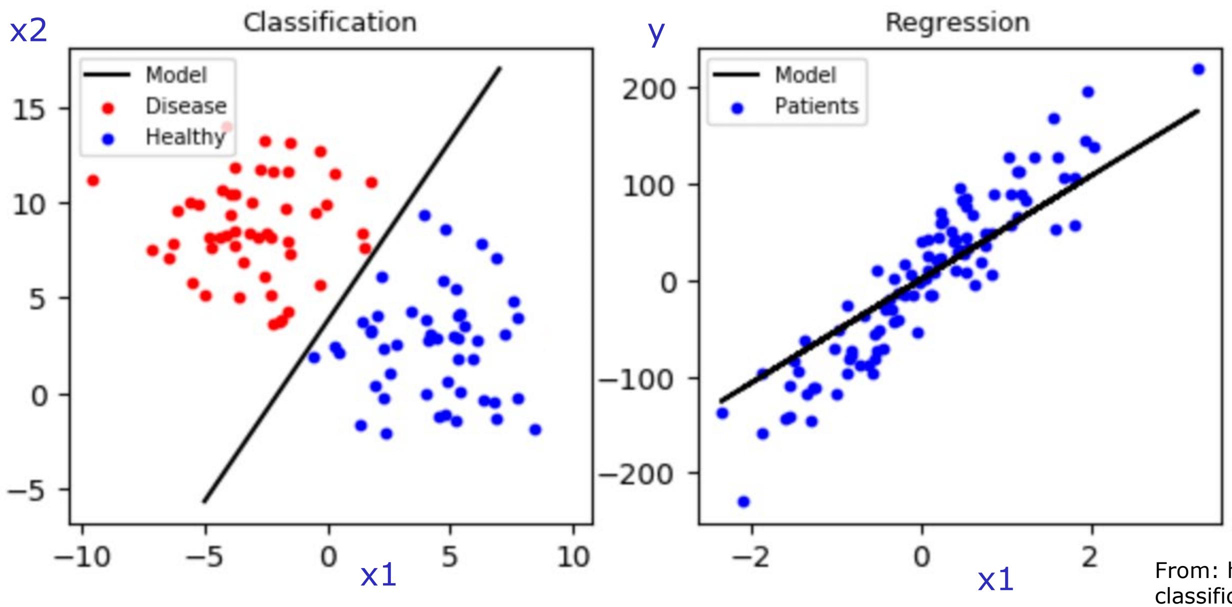
Machine learning with limited-size datasets

Michel Verleysen
Machine Learning Group
Université catholique de Louvain
Louvain-la-Neuve, Belgium
michel.verleysen@uclouvain.be

Machine learning and data analysis



From: https://mc.ai/between-classification-and-regression-how-to-know-which-is-which/

- Examples here:
  - Classification: 2-dimensional data
  - Regression: 1-dimensional data
- Machine learning is for *high*-dimensional data

High-dimensional learning and big data

- Is this high-dimensional? Big? Both?

GP	CMP	ATT	CMP%	YDS	AVG	TD	LNG	INT	FUM	QBR	RAT
3	9	16	56.3	65	4.06	0	16	1	2	--	39.8
2	6	15	40.0	46	3.07	0	16	0	1	7.8	48.2
2	20	28	71.4	218	7.79	1	43	0	0	78.0	106.0
16	341	536	63.6	4,038	7.53	28	71	13	6	64.4	93.8
16	350	541	64.7	4,434	8.20	30	83	7	8	70.7	103.2
15	312	475	65.7	3,922	8.26	28	86	11	2	69.2	101.2
15	343	502	68.3	4,643	9.25	45	93	6	4	84.5	122.5
16	371	552	67.2	4,295	7.78	39	73	8	5	71.2	108.0
9	193	290	66.6	2,536	8.75	17	83	6	3	60.6	104.9
16	341	520	65.6	4,381	8.43	38	80	5	7	78.3	112.2
16	347	572	60.7	3,821	6.68	31	65	8	8	60.3	92.7
16	401	610	65.7	4,428	7.26	40	66	7	4	73.8	104.2
6	128	193	66.3	1,385	7.18	13	72	3	1	62.6	103.2

High-dimensional learning and big data

- Is this high-dimensional? Big? Both?

High-dimensional

big data

GP	CMP	ATT	CMP%	YDS	AVG	TD	LNG	INT	FUM	QBR	RAT
3	9	16	56.3	65	4.06	0	16	1	2	--	39.8
2	6	15	40.0	46	3.07	0	16	0	1	7.8	48.2
2	20	28	71.4	218	7.79	1	43	0	0	78.0	106.0
16	341	536	63.6	4,038	7.53	28	71	13	6	64.4	93.8
16	350	541	64.7	4,434	8.20	30	83	7	8	70.7	103.2
15	312	475	65.7	3,922	8.26	28	86	11	2	69.2	101.2
15	343	502	68.3	4,643	9.25	45	93	6	4	84.5	122.5
16	371	552	67.2	4,295	7.78	39	73	8	5	71.2	108.0
9	193	290	66.6	2,536	8.75	17	83	6	3	60.6	104.9
16	341	520	65.6	4,381	8.43	38	80	5	7	78.3	112.2
16	347	572	60.7	3,821	6.68	31	65	8	8	60.3	92.7
16	401	610	65.7	4,428	7.26	40	66	7	4	73.8	104.2
6	128	193	66.3	1,385	7.18	13	72	3	1	62.6	103.2

High-dimensional learning and big data

- Is this high-dimensional? Big? Both?

High-dimensional, large p

big data, large n

GP	CMP	ATT	CMP%	YDS	AVG	TD	LNG	INT	FUM	QBR	RAT
3	9	16	56.3	65	4.06	0	16	1	2	--	39.8
2	6	15	40.0	46	3.07	0	16	0	1	7.8	48.2
2	20	28	71.4	218	7.79	1	43	0	0	78.0	106.0
16	341	536	63.6	4,038	7.53	28	71	13	6	64.4	93.8
16	350	541	64.7	4,434	8.20	30	83	7	8	70.7	103.2
15	312	475	65.7	3,922	8.26	28	86	11	2	69.2	101.2
15	343	502	68.3	4,643	9.25	45	93	6	4	84.5	122.5
16	371	552	67.2	4,295	7.78	39	73	8	5	71.2	108.0
9	193	290	66.6	2,536	8.75	17	83	6	3	60.6	104.9
16	341	520	65.6	4,381	8.43	38	80	5	7	78.3	112.2
16	347	572	60.7	3,821	6.68	31	65	8	8	60.3	92.7
16	401	610	65.7	4,428	7.26	40	66	7	4	73.8	104.2
6	128	193	66.3	1,385	7.18	13	72	3	1	62.6	103.2

High-dimensional learning and big data

- Is this high-dimensional? Big? Both?

High-dimensional, large *p*, complexity of the problem

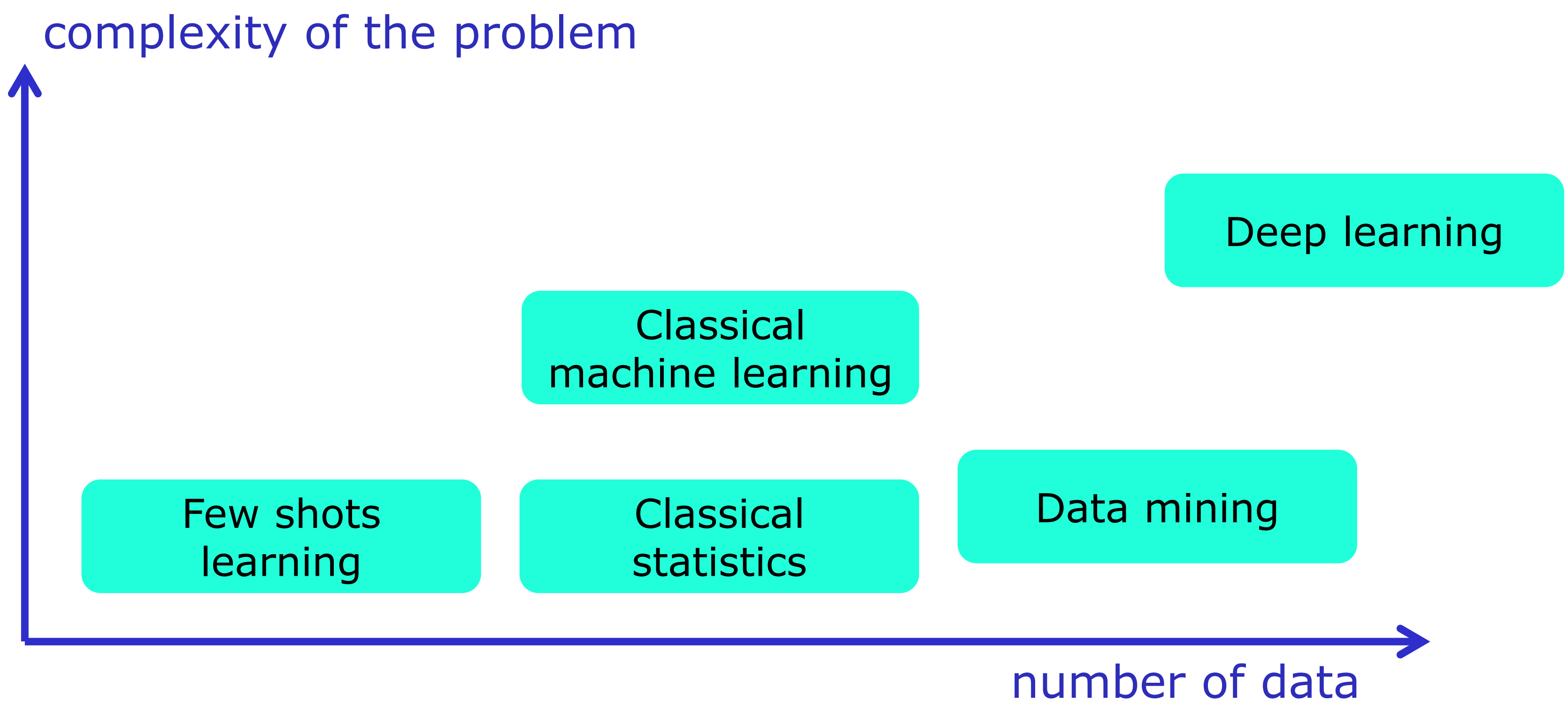
big data, large *n*, *abundance of information*

GP	CMP	ATT	CMP%	YDS	AVG	TD	LNG	INT	FUM	QBR	RAT
3	9	16	56.3	65	4.06	0	16	1	2	--	39.8
2	6	15	40.0	46	3.07	0	16	0	1	7.8	48.2
2	20	28	71.4	218	7.79	1	43	0	0	78.0	106.0
16	341	536	63.6	4,038	7.53	28	71	13	6	64.4	93.8
16	350	541	64.7	4,434	8.20	30	83	7	8	70.7	103.2
15	312	475	65.7	3,922	8.26	28	86	11	2	69.2	101.2
15	343	502	68.3	4,643	9.25	45	93	6	4	84.5	122.5
16	371	552	67.2	4,295	7.78	39	73	8	5	71.2	108.0
9	193	290	66.6	2,536	8.75	17	83	6	3	60.6	104.9
16	341	520	65.6	4,381	8.43	38	80	5	7	78.3	112.2
16	347	572	60.7	3,821	6.68	31	65	8	8	60.3	92.7
16	401	610	65.7	4,428	7.26	40	66	7	4	73.8	104.2
6	128	193	66.3	1,385	7.18	13	72	3	1	62.6	103.2

Small/large *p, n*

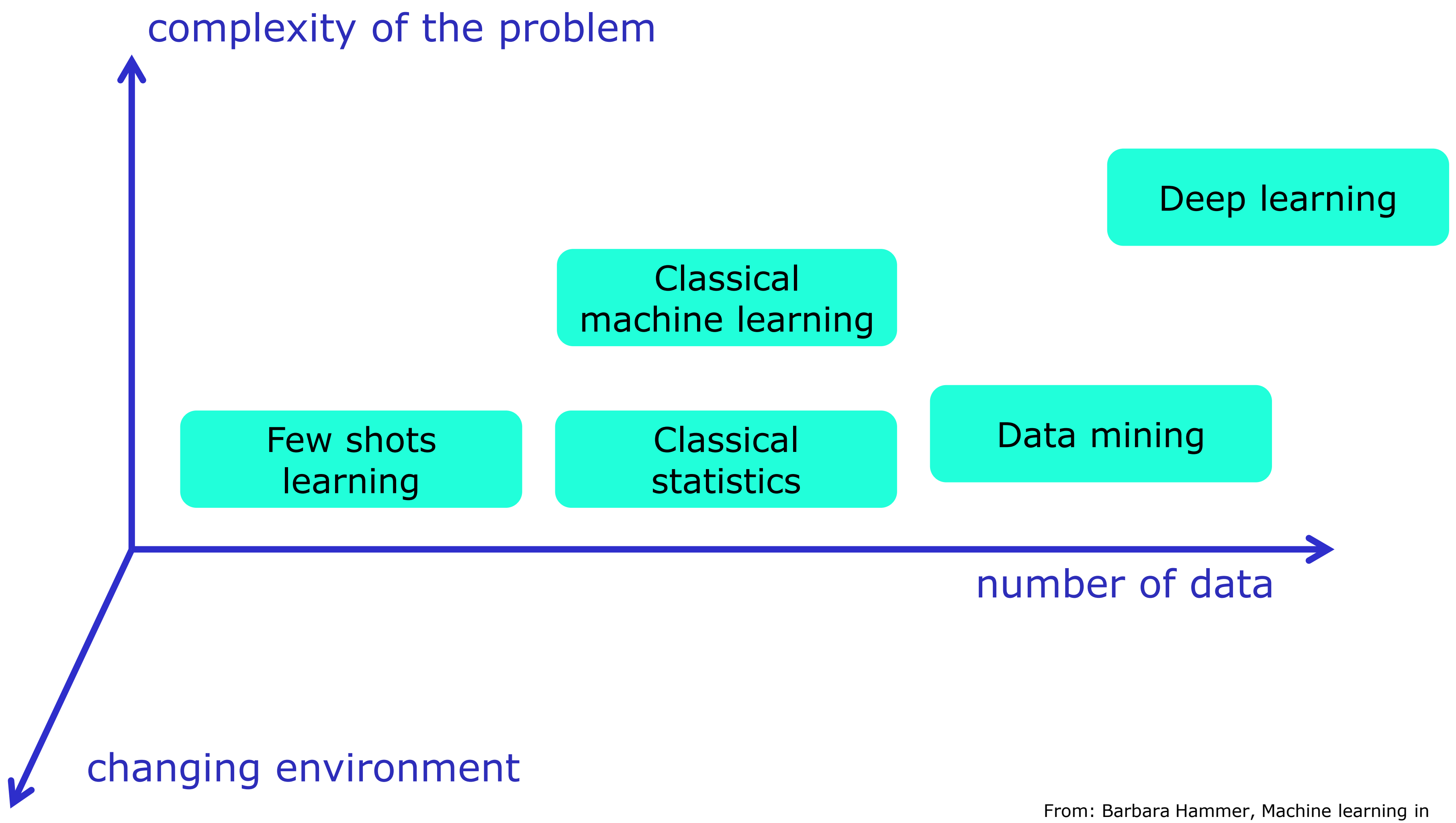
- Large *n* (many data)
  - Lot of information: <u>always</u> a good point (strong advantage)
  - Might lead to slow algorithms/computation (weak drawback)
  - Many applications do not come with many data! (medical records, time series, costly labels,…)
- Large *p* (many features)
  - Sensors and storage are easy and cheap
  - Curse of dimensionality
  - Useless features deteriorate learning

Machine learning



From: Barbara Hammer, Machine learning in the wild, plenary talk at WCCI 2020

Machine learning



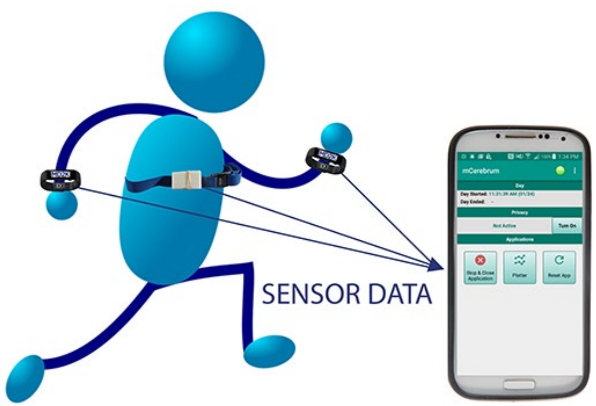
From: Barbara Hammer, Machine learning in the wild, plenary talk at WCCI 2020

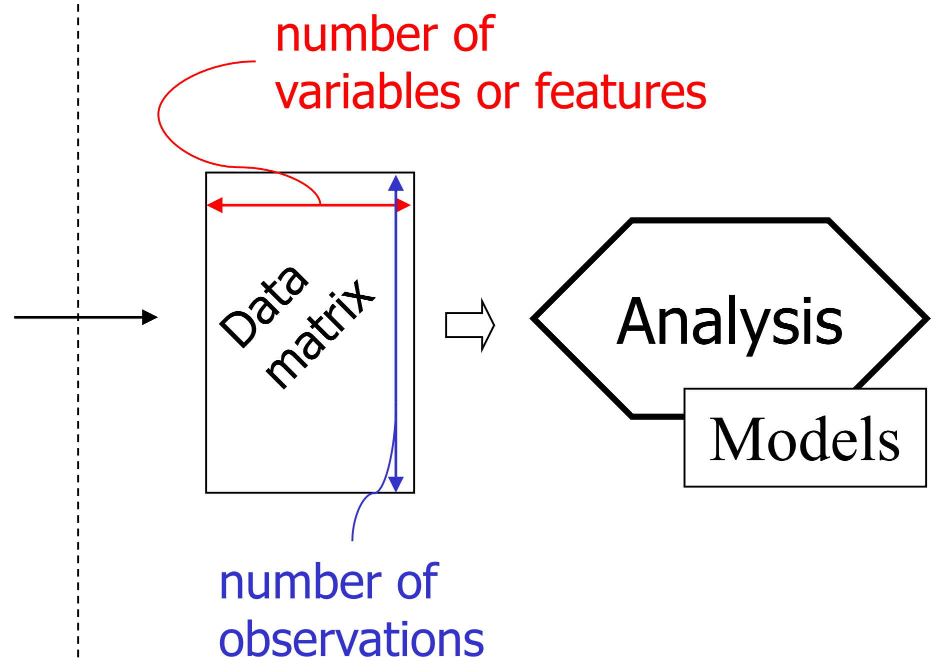
Outline

- High-dimensional data analysis
- The curse of dimensionality
- Feature selection
- Nonlinear dimensionality reduction
- Other small *n*, large *p* issues in machine learning
- Conclusion

High-dimensional data analysis

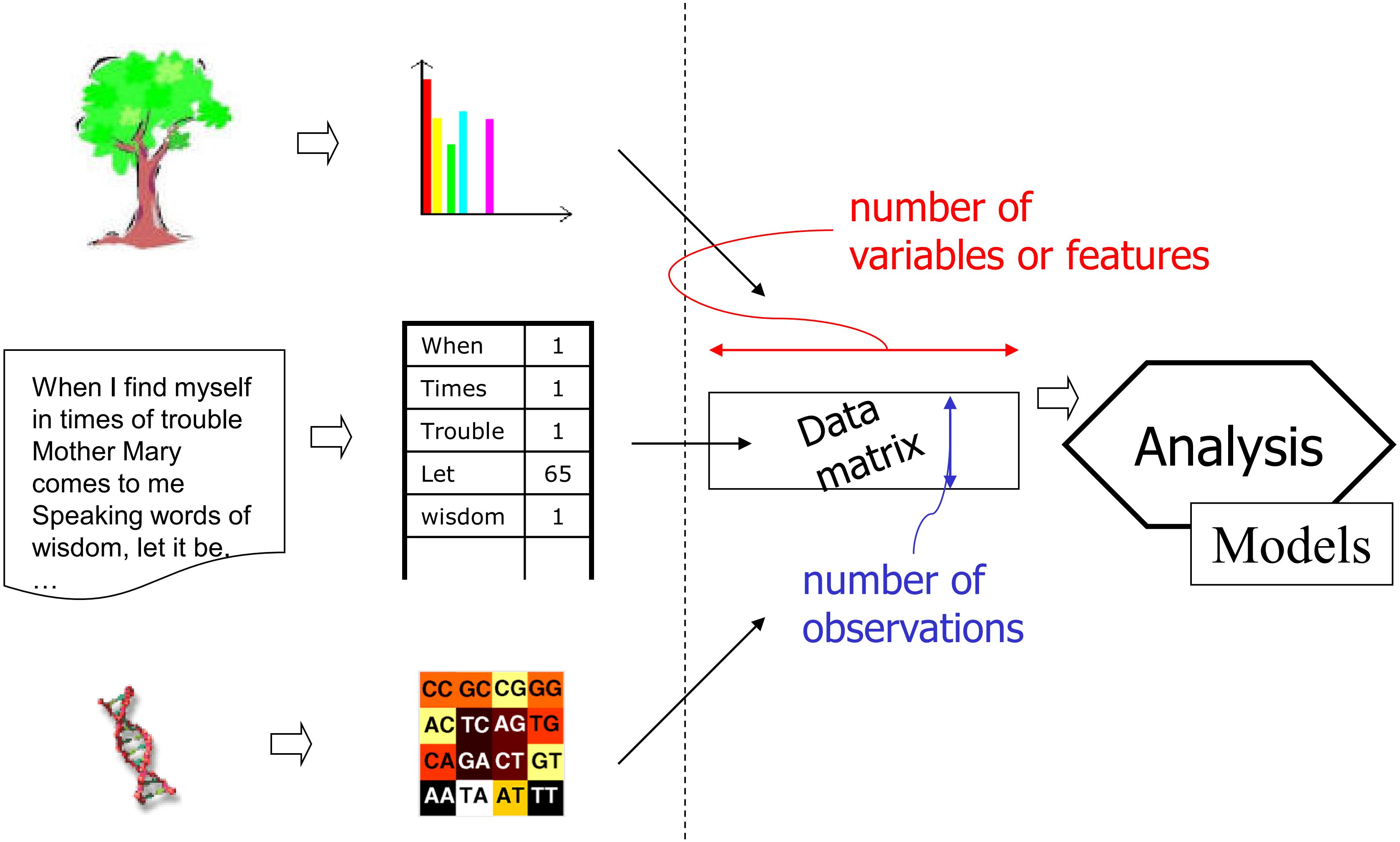
- Traditional data



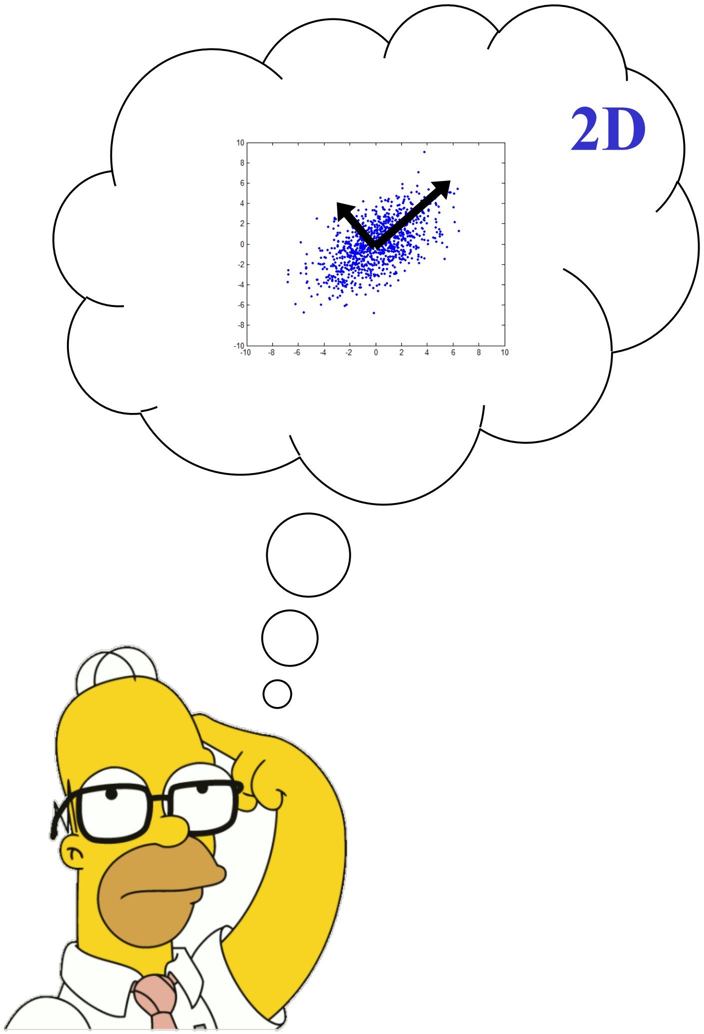


High-dimensional data analysis

- High-dimensional data

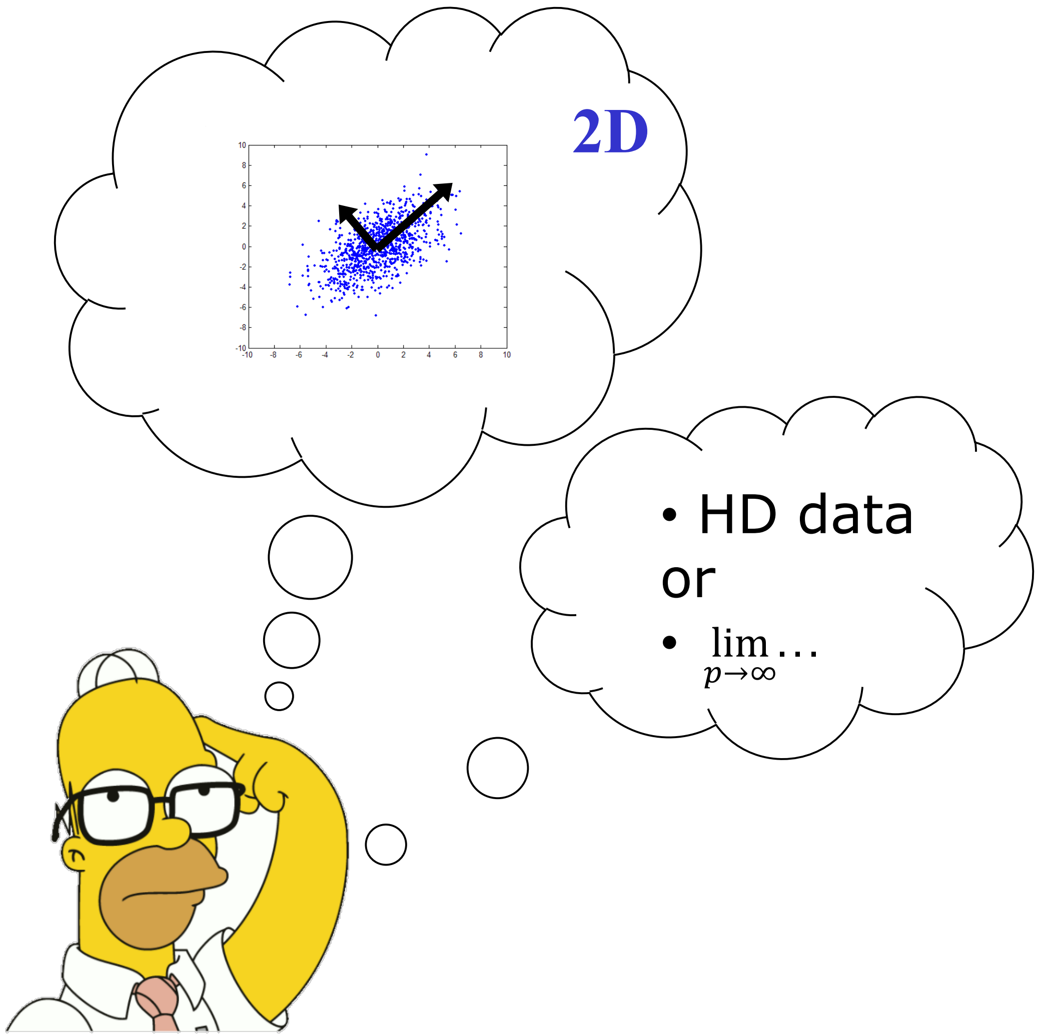


High-dimensional data and intuition



- Situations we can imagine, represent, draw
- Strong intuition of how the tools behave
- Consider cases where #data >> *p*

High-dimensional data and intuition

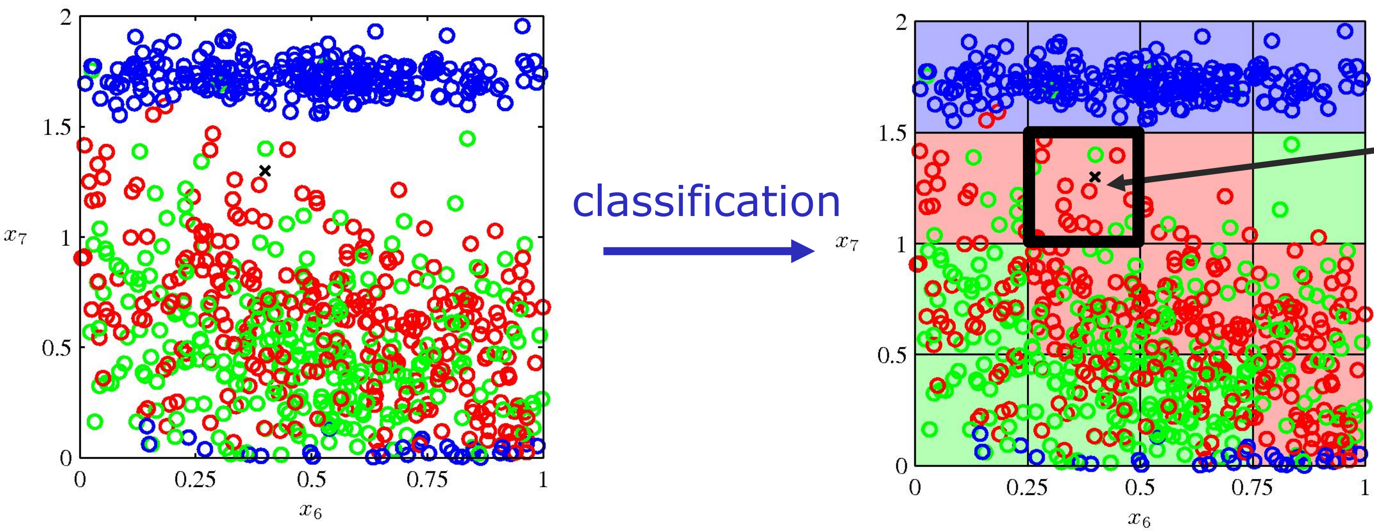


- ~~Situations we can imagine, represent, draw~~
- No representation

- ~~Strong intuition of how the tools behave~~
- No intuition

- ~~Consider cases where #observations >> e^p~~
- Often
  #observations << e^p

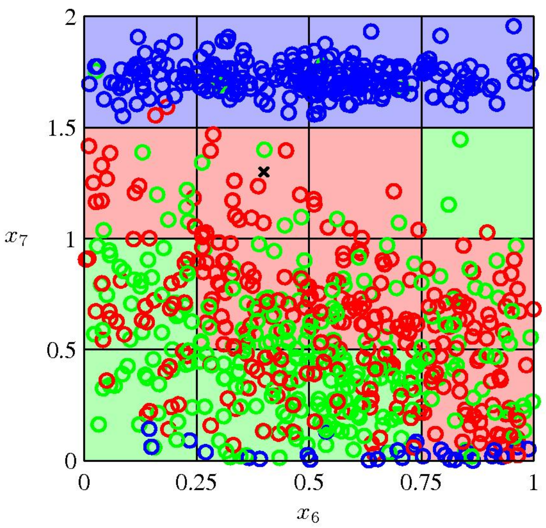
- Consider a simple classification algorithm (here in dimension 2=*p*):
  - split the space into boxes
  - decide the class of a box (here the colour) according to the majority class of the data in the box



this part of the data space is marked as *red* because it contains more red data than from any other class

Link between *p* and *n*

- What about the size and number of boxes ?
- Size: related to the accuracy of the algorithm
- Number: at *equal size* the number of boxes grows exponentially with the dimension *p* of the space
- And of course boxes must be populated with a sufficient number of data
- $\rightarrow$ Number of data *n* should grow exponentially with dimension *p*



Link between p and n

- In practice:

  Number of data n should grow exponentially with dimension p

  is not realistic…

Link between *p* and *n*

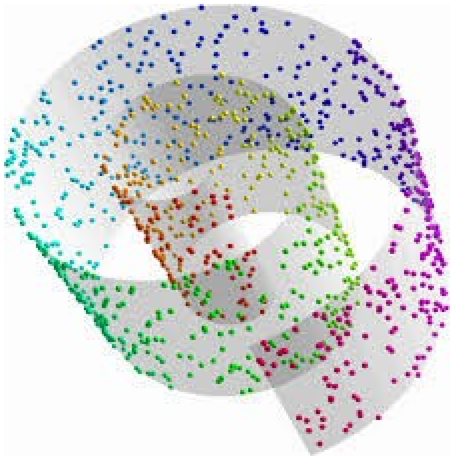
- In practice:

  Number of data *n* should grow exponentially with dimension *p*

  is not realistic…

- Hopefully real data have a structure: this is not really 3-D data

- Data without structure do not contain information!



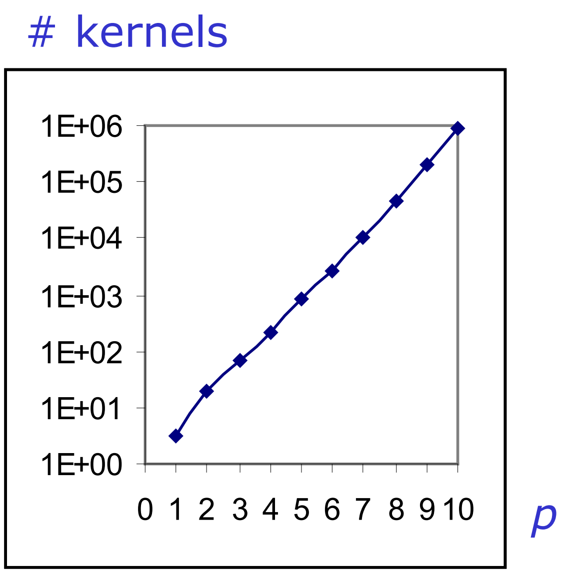
From: https://www.kdnuggets.com/2015/01/yoshua-bengio-unsupervised-learning-robust-adversarial-distortions.html

Outline

- High-dimensional data analysis
- The curse of dimensionality
- Feature selection
- Nonlinear dimensionality reduction
- Other small *n*, large *p* issues in machine learning
- Conclusion

Curse of dimensionality

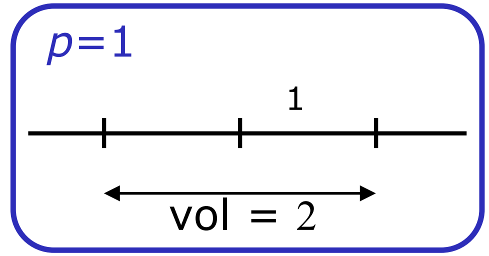
- Example: Silverman (1986)
  Number of Gaussian kernels necessary to approximate a (Gaussian) distribution in dimension *p*

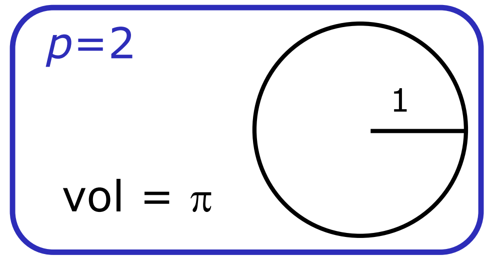


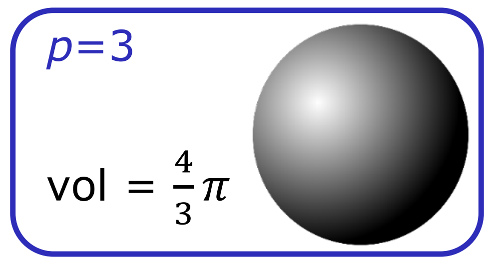
Surprizing facts: sphere

- Volume of “sphere” of constant radius (=1) in dimension *p*

- A “sphere” is: a segment (*p*=1)
  a circle (*p*=2)
  a sphere (*p*=3)
  a hypershere (*p*>3)



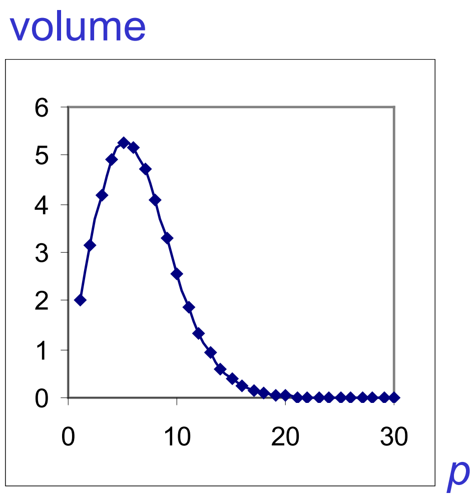




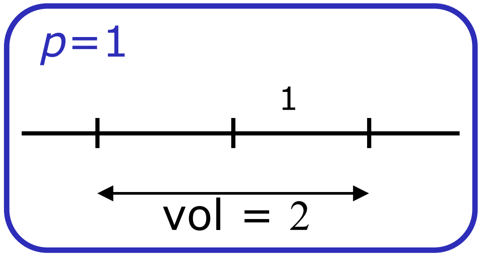
Surprizing facts: sphere

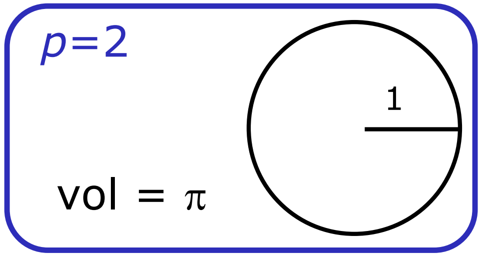
- Volume of "sphere" of constant radius (=1) in dimension *p*

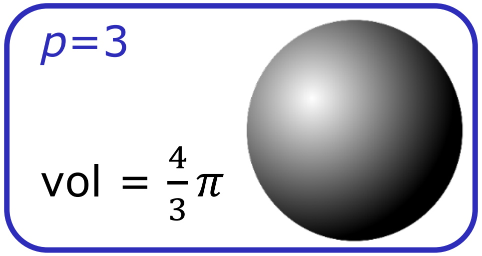
- A "sphere" is: a segment (*p*=1)
  a circle (*p*=2)
  a sphere (*p*=3)
  a hypershere (*p*>3)



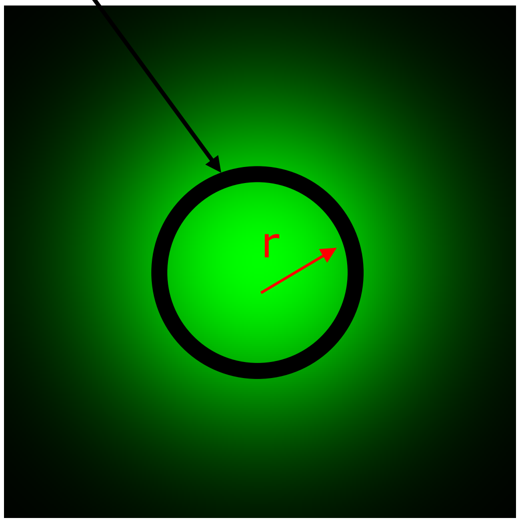
$$\mathrm{vol}(p) = \frac{\pi^{p/2}}{\Gamma(p/2+1)} r^p$$

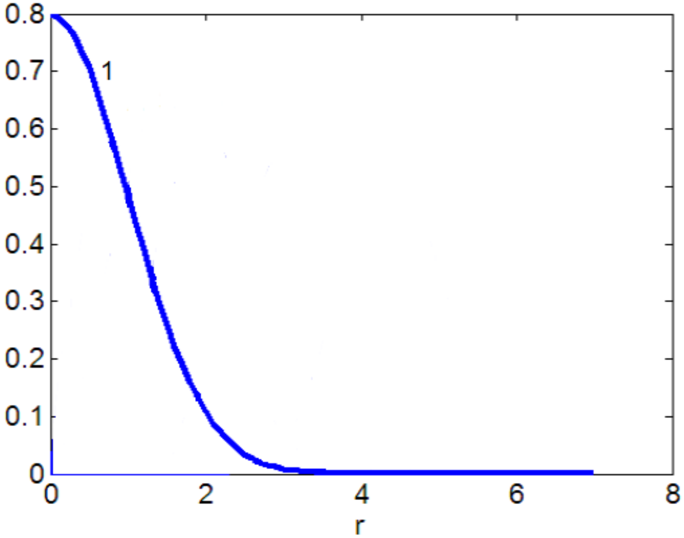






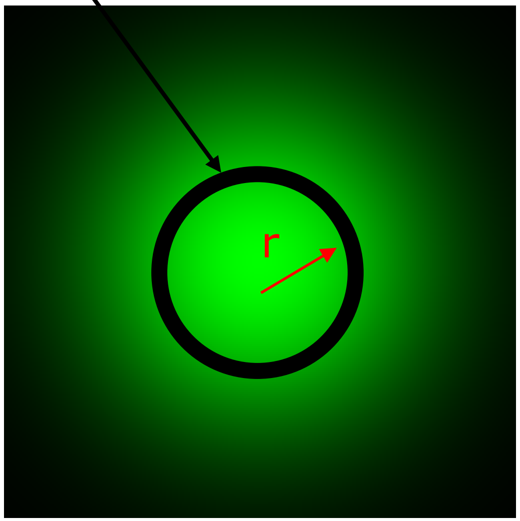
- Another view of high-DIM Gaussian distributions:
  - Probability to find a point at distance *r* from the center of a DIM-dimensional multinormal distribution

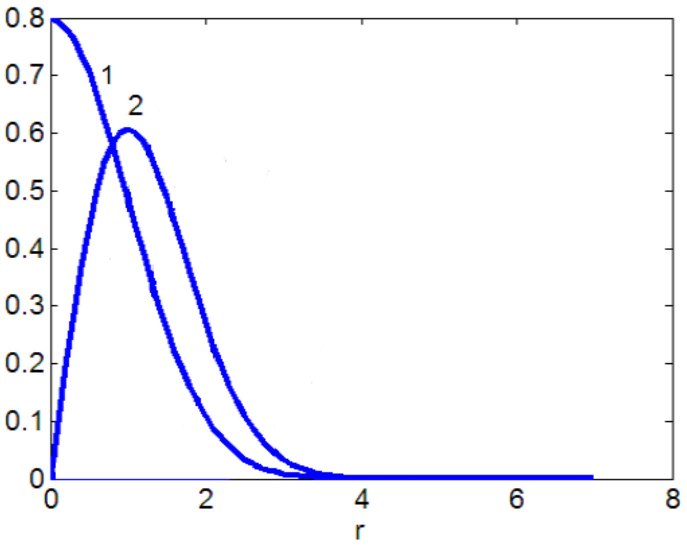




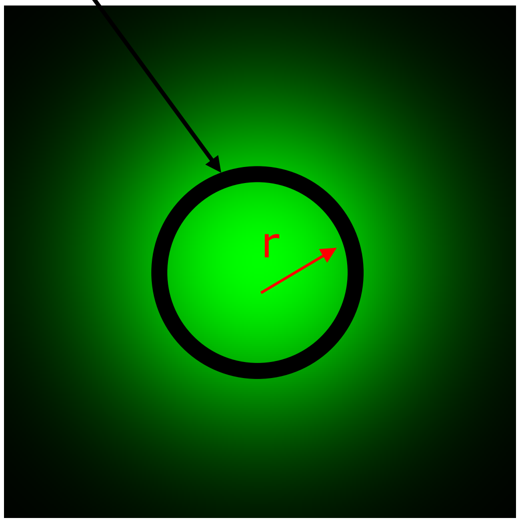
Surprizing facts: Gaussians

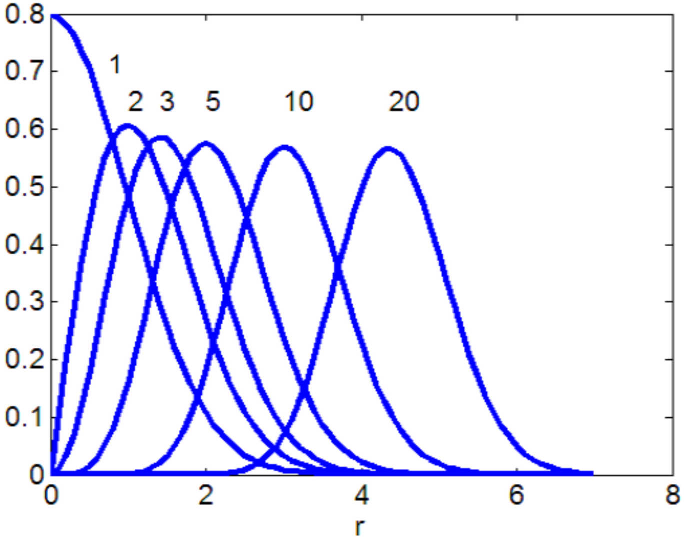
- Another view of high-DIM Gaussian distributions:
  - Probability to find a point at distance *r* from the center of a DIM-dimensional multinormal distribution





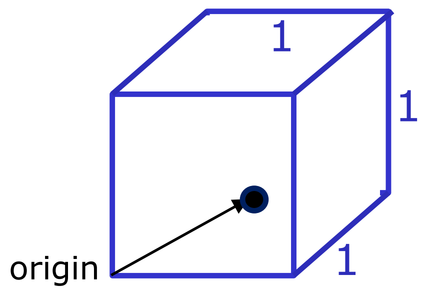
- Another view of high-DIM Gaussian distributions:
  - Probability to find a point at distance *r* from the center of a DIM-dimensional multinormal distribution





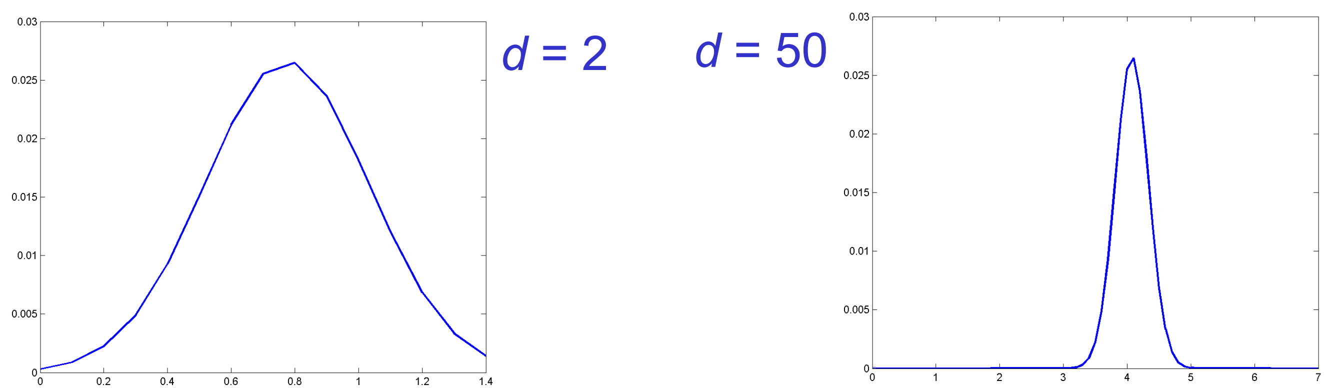
Concentration of the Euclidean norm

- Let's compute the norm of random vectors in a cube



Data (not shown):
i.i.d. components in [0,1]

- Distribution of the norm of random vectors

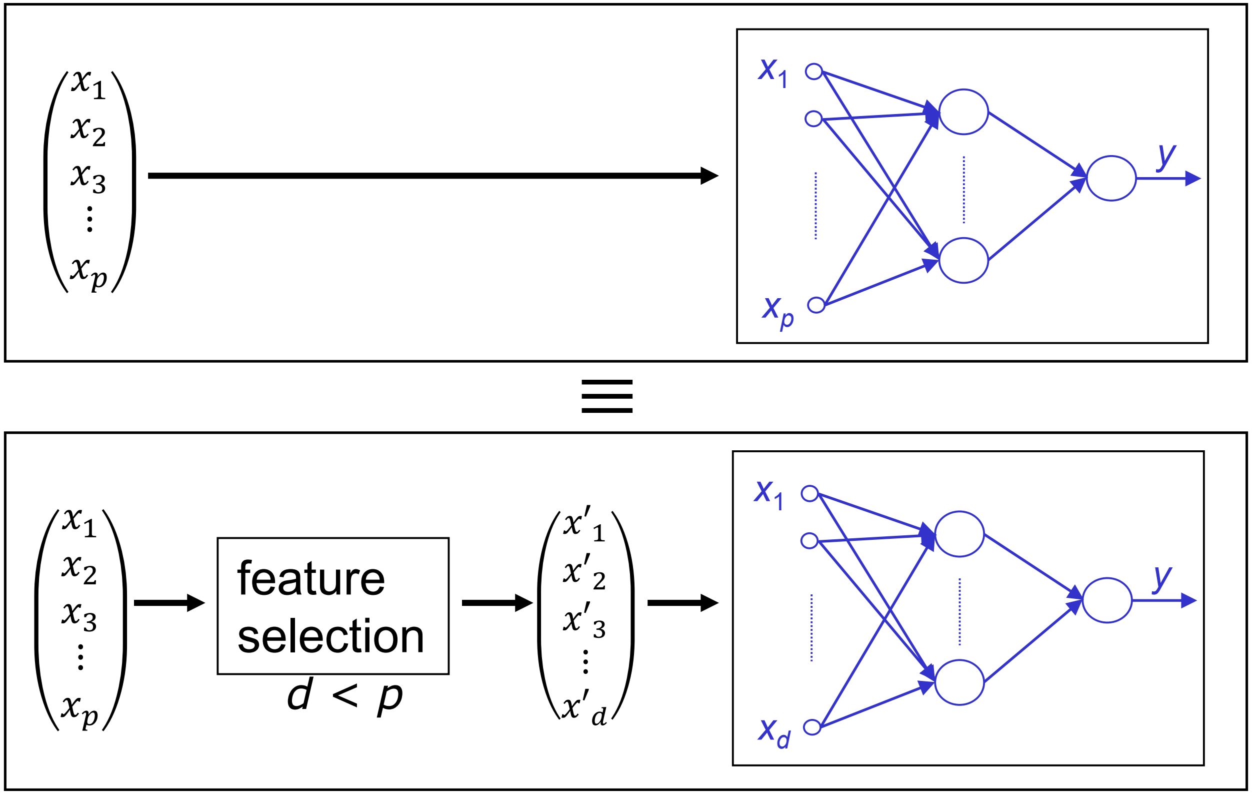


- Norms concentrate around their expectation
- They don't discriminate anymore ! (remember that many machine learning algorithms are based on distances between data, or norms)

Outline

- High-dimensional data analysis
- The curse of dimensionality
- Feature selection
- Nonlinear dimensionality reduction
- Other small *n*, large *p* issues in machine learning
- Conclusion

Feature selection reduces *p*, keeps *n*



Feature selection: the x'_i features are *among* the original set $\{x_j\}$

Original features are preserved → interpretability as well

Feature selection reduces p, keeps n

- Feature selection keeps some features among the original ones:

GP	CMP	ATT	CMP%	YDS	AVG	TD	LNG	INT	FUM	QBR	RAT
3	9	16	56.3	65	4.06	0	16	1	2	--	39.8
2	6	15	40.0	46	3.07	0	16	0	1	7.8	48.2
2	20	28	71.4	218	7.79	1	43	0	0	78.0	106.0
16	341	536	63.6	4,038	7.53	28	71	13	6	64.4	93.8
16	350	541	64.7	4,434	8.20	30	83	7	8	70.7	103.2
15	312	475	65.7	3,922	8.26	28	86	11	2	69.2	101.2
15	343	502	68.3	4,643	9.25	45	93	6	4	84.5	122.5
16	371	552	67.2	4,295	7.78	39	73	8	5	71.2	108.0
9	193	290	66.6	2,536	8.75	17	83	6	3	60.6	104.9
16	341	520	65.6	4,381	8.43	38	80	5	7	78.3	112.2
16	347	572	60.7	3,821	6.68	31	65	8	8	60.3	92.7
16	401	610	65.7	4,428	7.26	40	66	7	4	73.8	104.2
6	128	193	66.3	1,385	7.18	13	72	3	1	62.6	103.2

Why feature selection ?

Three good reasons for feature selection:

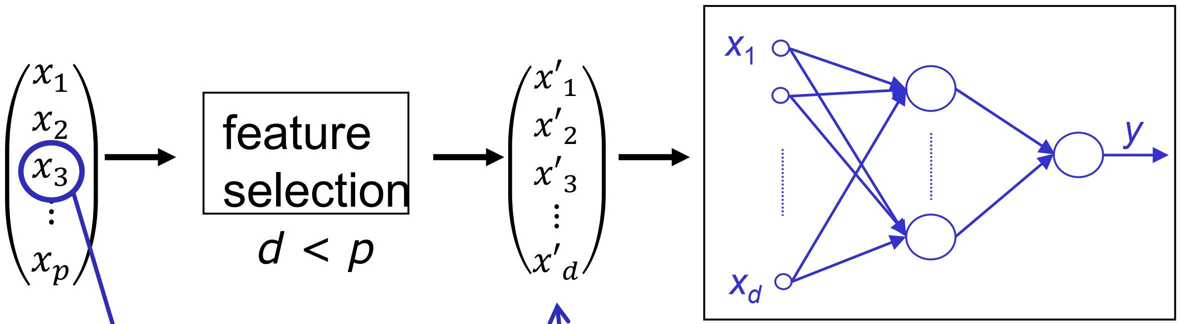
1. improve the models
   - reduces p, keeps n: good for learning (whatever is the model)

2. explain the features
   - many applications require insights about the features (biomarkers, costly sensors, …)

3. visualize data
   - visualization is mostly restricted to 2-D or 3-D data
   - visualization is an essential part of a data analysis process

Univariate versus multivariate feature selection

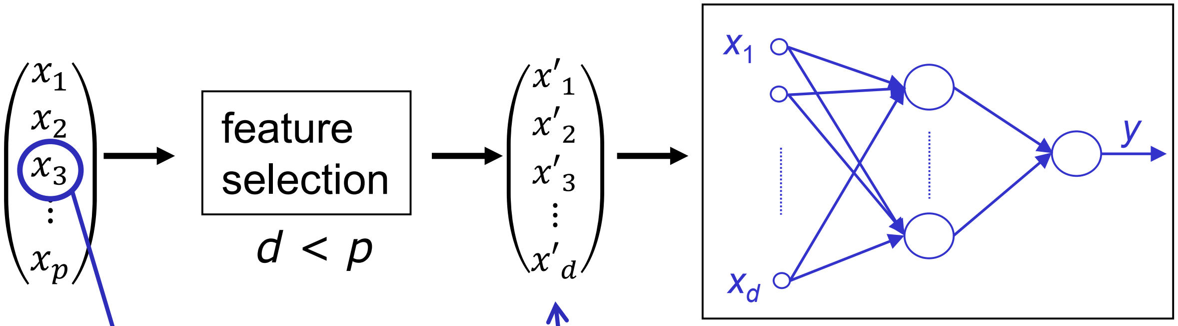
- Univariate FS:
  each original feature is evaluated independently from the other ones



should this feature
be included in the set ?

Univariate versus multivariate feature selection

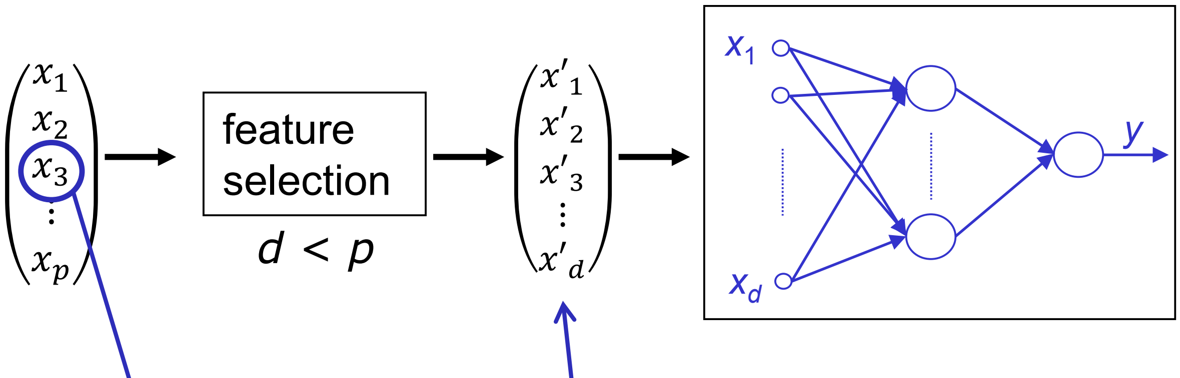
- Multivariate FS:
  each original feature is evaluated conditionally to other ones



should this feature
be included in the set,
given that other features have
already been selected ?

Univariate versus multivariate feature selection

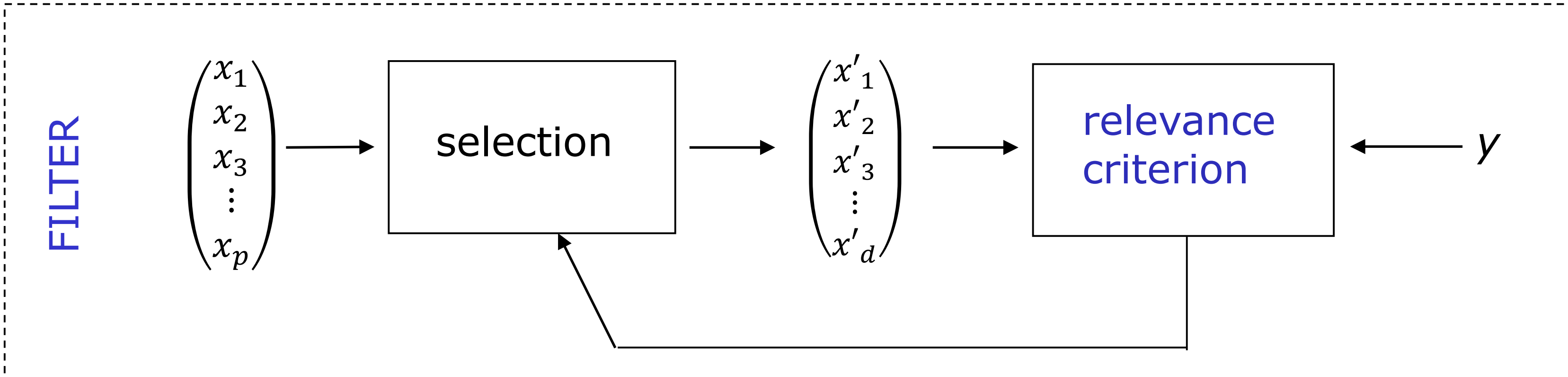
- Multivariate FS:
  each original feature is evaluated conditionally to other ones



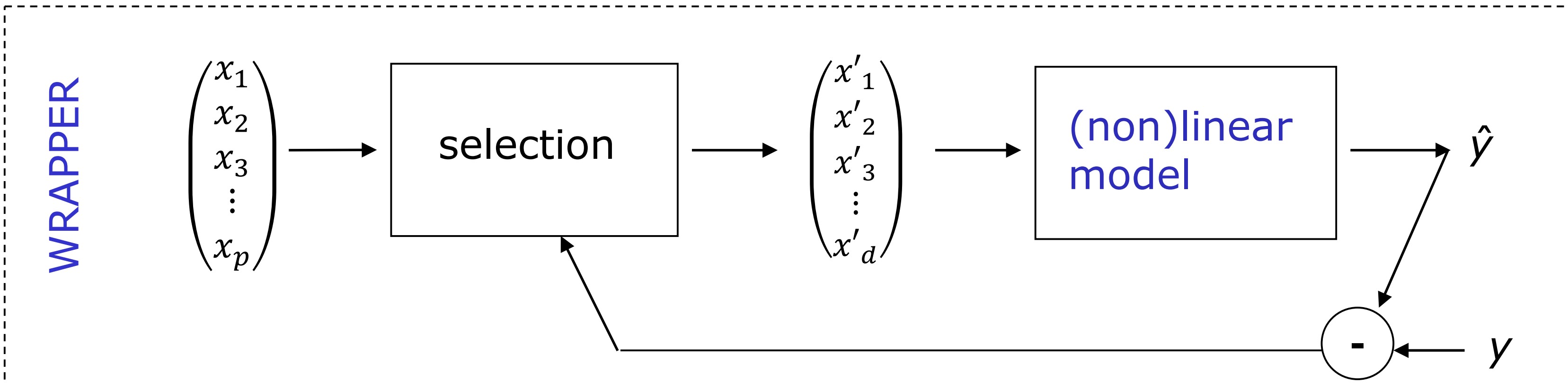
should this feature
be included in the set,
given that other features have
already been selected ?

- Multivariate FS makes it possible to detect features that contribute together to y, but not individually (ex: $y = x_1 \oplus x_2$)

Filters versus wrappers feature selection



Not the final quality criterion !



Many (non)linear models to design!

Filters/wrappers – univariate/multivariate

<table>
<tr><th></th><th>Filters</th><th>Wrappers</th></tr>
<tr><td rowspan="1">Univariate</td><td colspan="2">Use only if p huge
(so huge that any other approach won’t work)</td></tr>
<tr><td rowspan="2">Multivariate</td><td colspan="2">Only for moderate <u>final</u> p
(otherwise the feature selection itself is difficult because of the curse of dimensionality)</td></tr>
<tr><td>• Model agnostic (ideal for feature interpretation)
• Moderate computational load</td><td>• Ideal performances for specific model
• High computational load</td></tr>
</table>

Common mistake in literature: filters ≠ univariate !

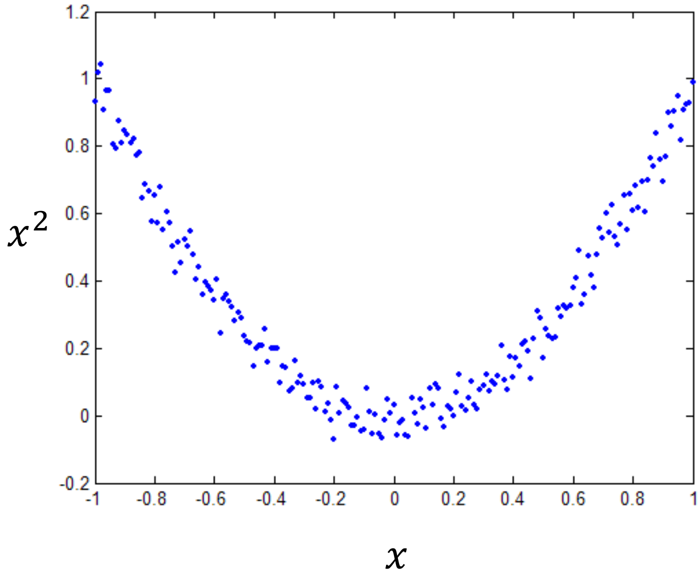
Multivariate feature selection

The two ingredients of feature selection:

1. a criterion to evaluate subsets of features
   - wrappers: the model itself
   - filters: a.o. mutual information
2. a subset selection method
   - $2^p - 1$ possible subsets: intractable if p is huge
   - need to try only some of the $2^p - 1$ possible subsets

1. criterion

- Correlation is really a bad idea…
  - it does not see nonlinear relations
    (ex.: correlation between x and x^2 is zero…)



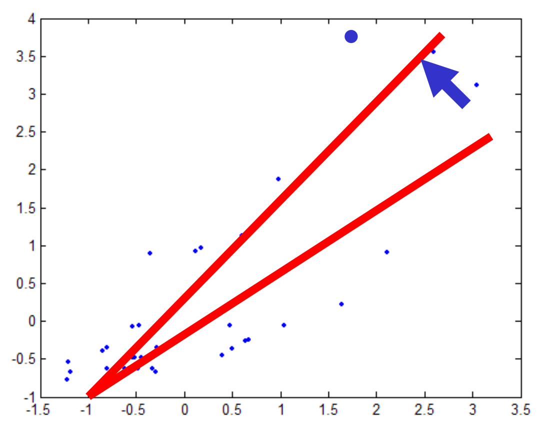
- There is a clear relation
- However the correlation is 0

1. criterion

- Correlation is really a bad idea…
  - it does not see nonlinear relations
    (ex.: correlation between x and x^2 is zero…)
  - it is not multivariate
    (what is the correlation between y and $\{x_1, x_7, x_{13}\}$?)

1. criterion

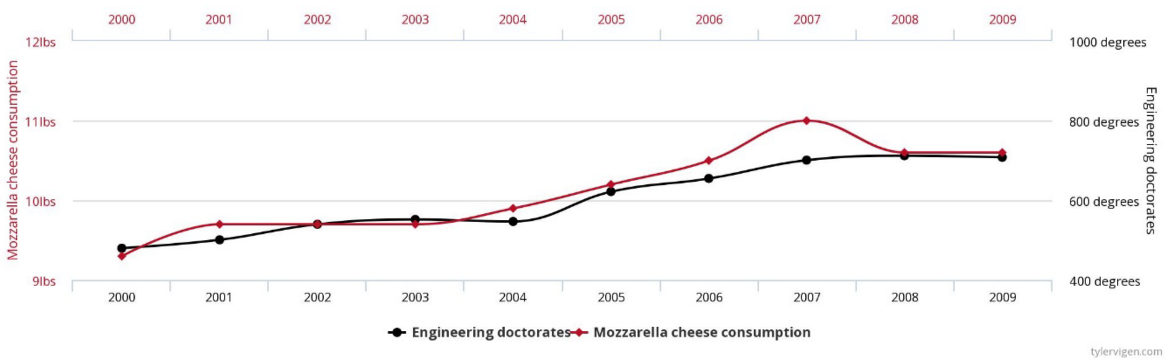
- Correlation is really a bad idea…
  - it does not see nonlinear relations
    (ex.: correlation between x and x^2 is zero…)
  - it is not multivariate
    (what is the correlation between y and $\{x_1, x_7, x_{13}\}$?)
  - it is extremely sensitive to outliers
    (equivalent to linear regression, based on <u>squared</u> distances or errors)



1. criterion

- Correlation is really a bad idea…
  - it does not see nonlinear relations
    (ex.: correlation between x and x^2 is zero…)
  - it is not multivariate
    (what is the correlation between y and $\{x_1, x_7, x_{13}\}$?)
  - it is extremely sensitive to outliers
    (equivalent to linear regression, based on <u>squared</u> distances or errors)
  - it is not causal
    (ex.: the high correlation between the number of murders and churches in US towns is due to…the size of the town)

- Correlation is really a bad idea…
  - it does not see nonlinear relations
    (ex.: correlation between x and x^2 is zero…)
  - it is not multivariate
    (what is the correlation between y and $\{x_1, x_7, x_{13}\}$?)
  - it is extremely sensitive to outliers
    (equivalent to linear regression, based on squared distances or errors)
  - it is not causal
    (ex.: the high correlation between the number of murders and churches in US towns is due to…the size of the town)
    (other ex., more funny: civil engineering doctorate awarded wrt consumption of mozzarella cheese:)



1. criterion

- Correlation is really a bad idea…
  - it does not see nonlinear relations
    (ex: correlation between x and x^2 is zero…)
  - it is not multivariate
    (what is the correlation between y and $\{x_1, x_7, x_{13}\}$?)
  - it is extremely sensitive to outliers
    (equivalent to linear regression, based on squared distances or errors)
  - it is not causal
    (ex: the high correlation between the number of murders and churches in US towns is due to…the size of the town)

- Mutual information is a much better idea !
  - it solves the three first issues
  - however it must be estimated from the data, which is itself a HD problem
    (but here HD relates to the final set of features, not the initial one)

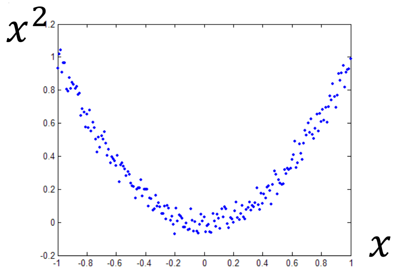
1. criterion: mutual information

- Mutual information in a nutshell

$$I(y\,;x) = H(y) - H(y|x) = H(x) - H(x|y)$$

- Difference between the entropy of y (uncertainty on the prediction) and the entropy of y when x is known
- Measures how much x gives information about y

- Example:
  - x and z uniformly distributed over [-1 1]
  - z is independent from x



	$x^2; x^2$	$x; x^2$	$z; x^2$
Correlation	1	0.05	0.05
Mutual information	2.26	1.20	0.01

no difference

clear difference

1. criterion: mutual information

$$I(y\,;x) = H(y) - H(y|x) = H(x) - H(x|y)$$

- x can be *a set of features* (multivariate criterion!)

- The difficulty is in the *estimation* of $I\,(y;x)$:
  - The estimators also suffer from the curse of dimensionality!
  - The subset selection method should be limited to the *final* number of features, not the *initial* number
    (ex.: bioinformatics, selection of 50 genes among 50.000)

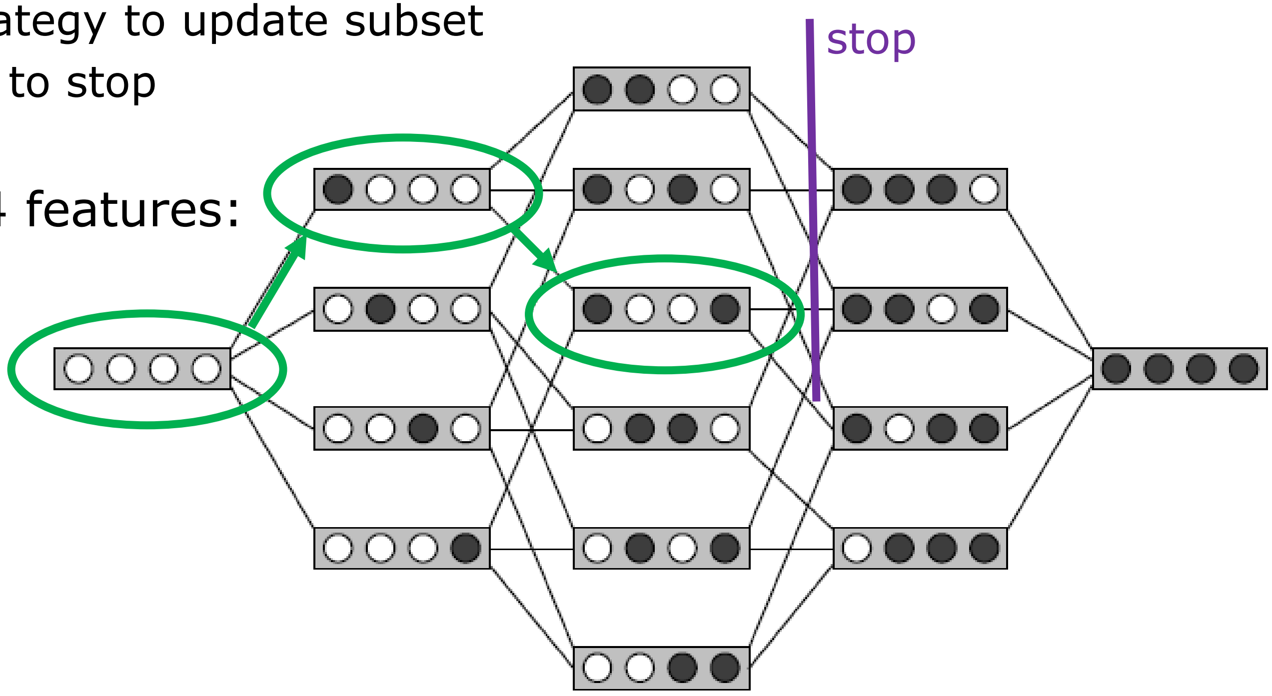
- Mutual information is not limited to a fixed interval such as $[-1,1]$
  - it is limited by the entropies of x and y, which are generally unknown.

2. feature subset selection method

- In theory: just try $2^p - 1$ subsets and evaluate them...
  - Evaluation: mutual information (filters), or model itself (wrapper)
  - just imagine for $p = 200$ ☺
- In practice: greedy procedure
  - Define an intial subset
  - Choose a strategy to update subset
  - Decide when to stop

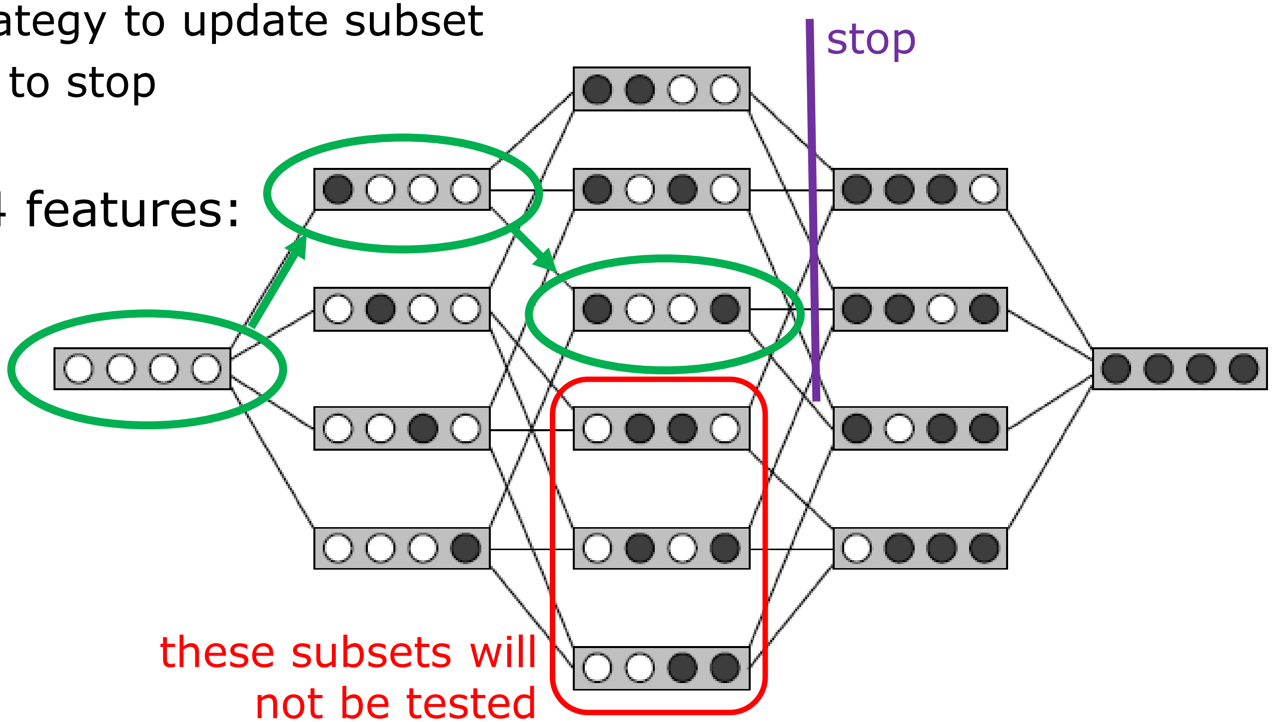
2. feature subset selection method

- In theory: just try 2^p-1 subsets and evaluate them...
  - Evaluation: mutual information (filters), or model itself (wrapper)
  - just imagine for d=200 ☺
- In practice: greedy procedure
  - Define an intial subset
  - Choose a strategy to update subset
  - Decide when to stop
- Example with 4 features:



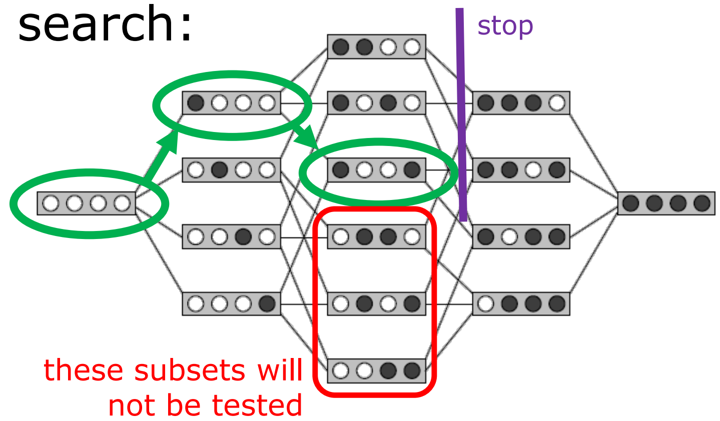
2. feature subset selection method

- In theory: just try 2^p-1 subsets and evaluate them...
  - Evaluation: mutual information (filters), or model itself (wrapper)
  - just imagine for d=200 ☺
- In practice: greedy procedure
  - Define an intial subset
  - Choose a strategy to update subset
  - Decide when to stop
- Example with 4 features:



2. feature subset selection method

- This is the forward search:



- Backward search: start from the full set of features
  - better to detect dependencies
  - very bad is initial *p* is too big (estimation problem)

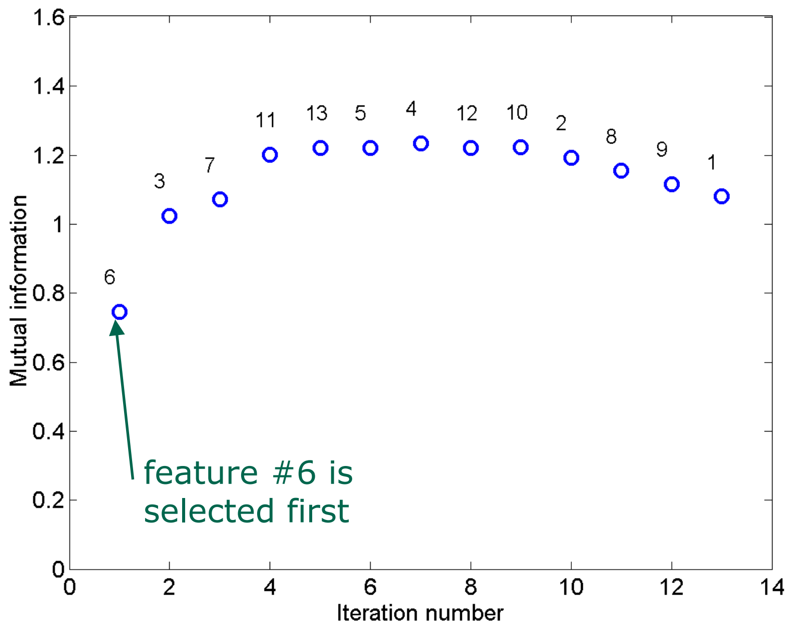
- Other search methods:
  - forward-backward
  - MRMR (limited to 2-D estimations, pairs of features only)
  - genetic algorithms
  - …

Feature selection example

- Dataset origin: StatLib library, Carnegie Mellon Univ.
- Concerns housing values in suburbs of Boston
- Attributes:
  1. CRIM per capita crime rate by town
  2. ZN proportion of residential land zoned for lots over 25,000 sq.ft.
  3. INDUS proportion of non-retail business acres per town
  4. CHAS Charles River dummy variable (= 1 if tract bounds river, 0 otherw.)
  5. NOX nitric oxides concentration (parts per 10 million)
  6. RM average number of rooms per dwelling
  7. AGE proportion of owner-occupied units built prior to 1940
  8. DIS weighted distances to five Boston employment centres
  9. RAD index of accessibility to radial highways
  10. TAX full-value property-tax rate per $10,000
  11. PTRATIO pupil-teacher ratio by town
  12. B 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
  13. LSTAT % lower status of the population
  14. MEDV Median value of owner-occupied homes in $1000's

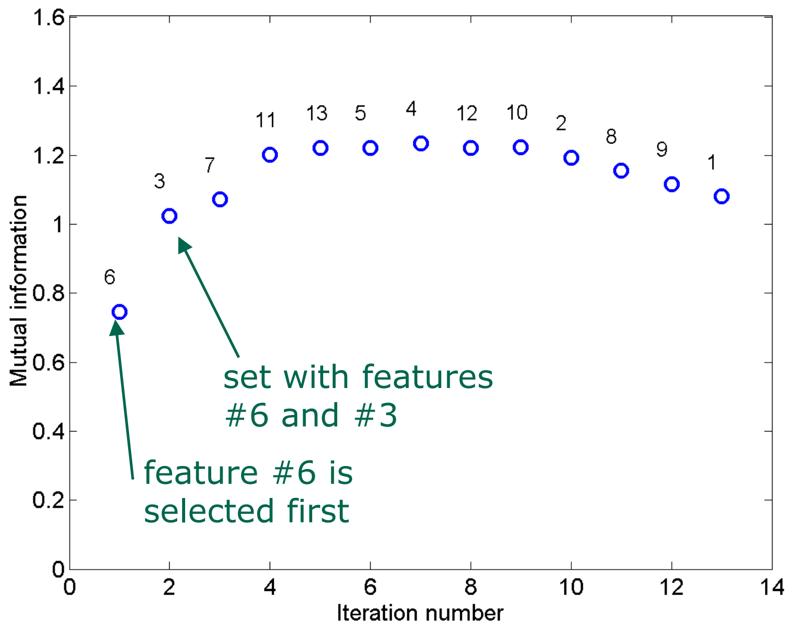
Feature selection example

- Forward selection with MI:



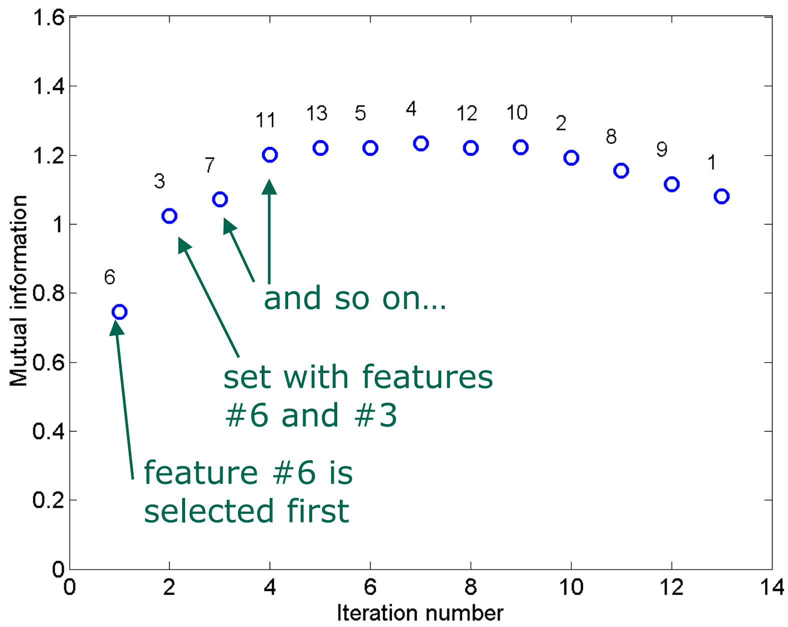
Feature selection example

- Forward selection with MI:



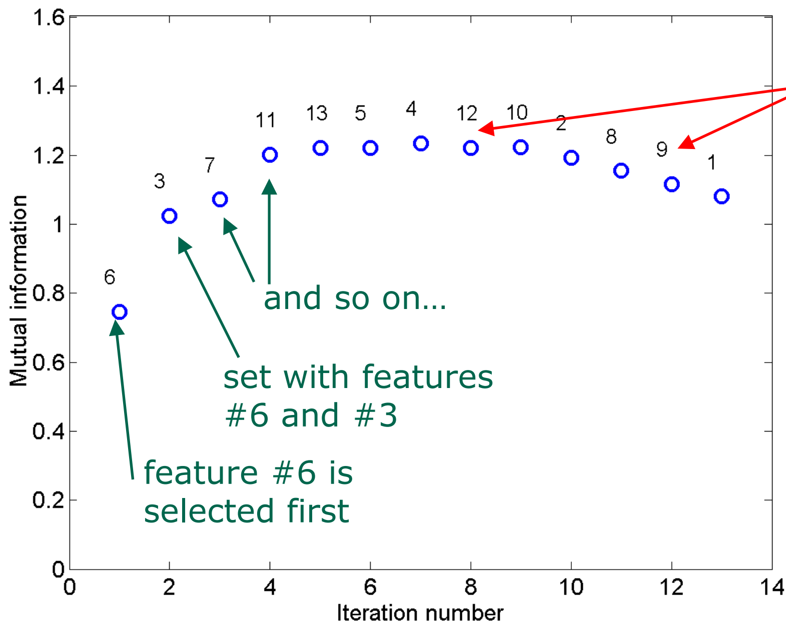
Feature selection example

- Forward selection with MI:



Feature selection example

- Forward selection with MI:



in theory: should never decrease

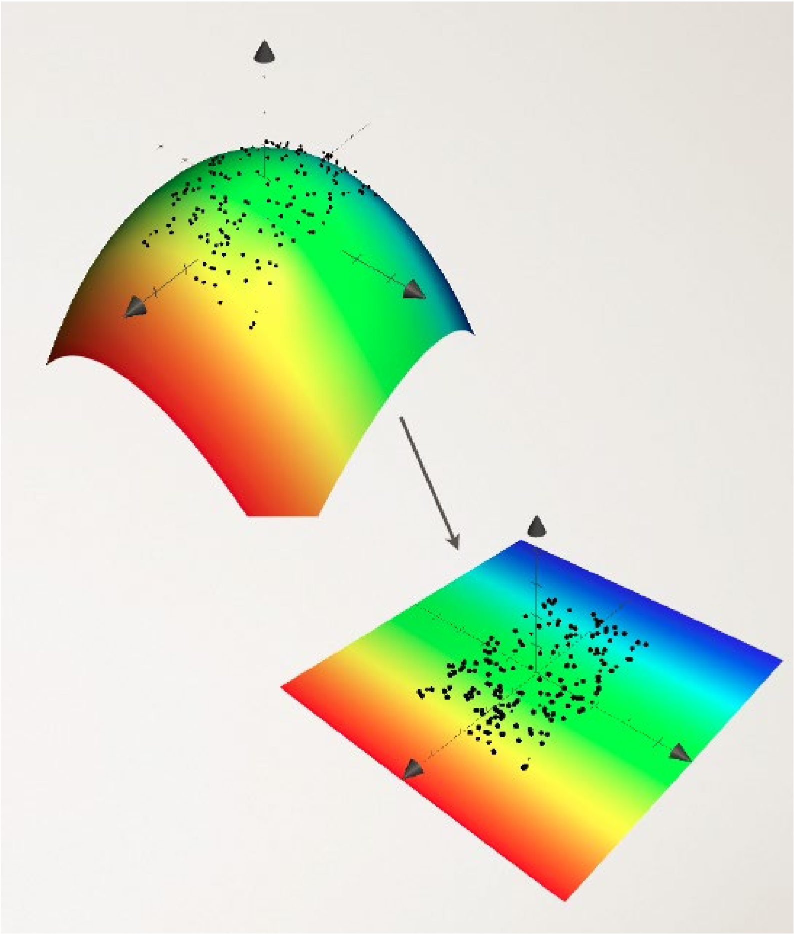
in practice: limitation of estimator

Outline

- High-dimensional data analysis
- The curse of dimensionality
- Feature selection
- Nonlinear dimensionality reduction
- Other small n, large p issues in machine learning
- Conclusion

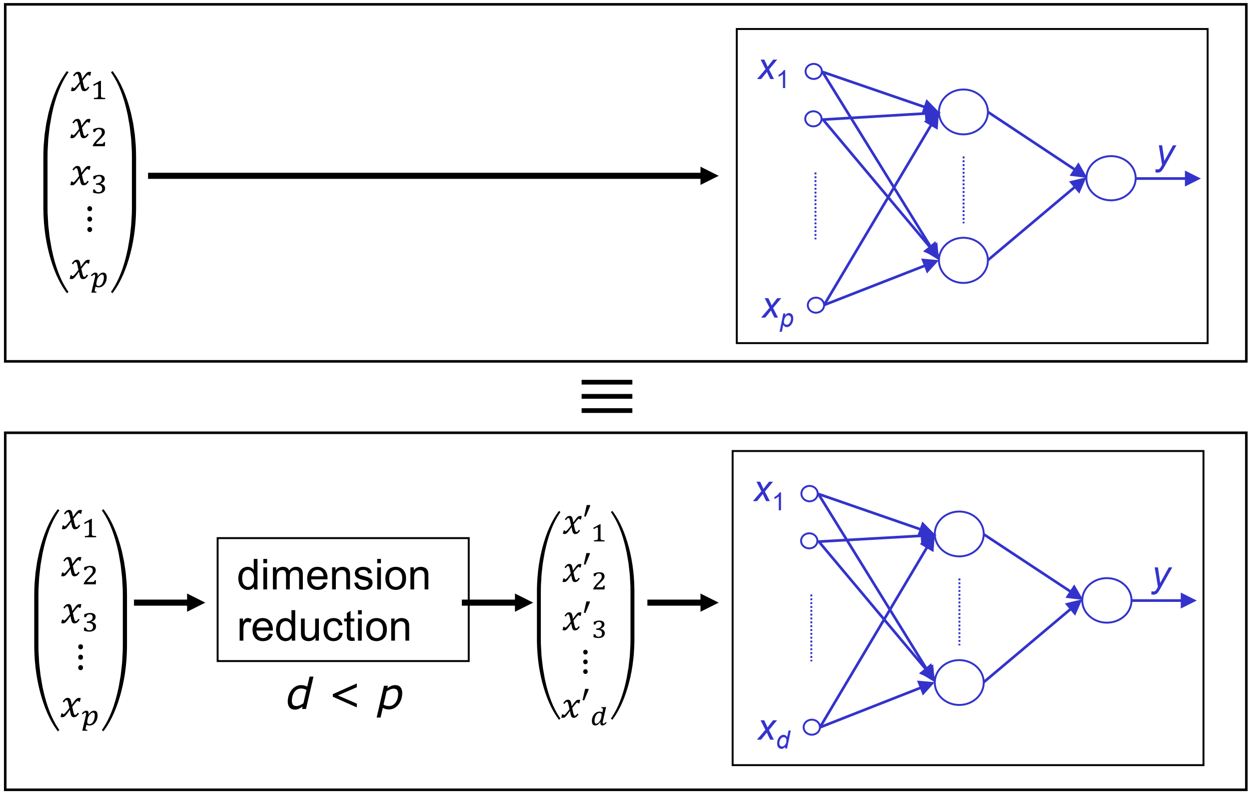
Nonlinear dimensionality reduction

- Think flattening a 3D surface to a 2D image
  - Simple projection
  - Preserving some particular quantity of interest locally
  - Preserving some global property of the surface



From Ryan B. Harvey, Methods of manifold learning for dimension reduction of large datasets, PhD oral exam, 2010

Nonlinear dimensionality reduction



Dimension reduction: features x'_i are *built from* the original set $\{x_j\}$
(Feature selection: features x'_i are *among* the original set $\{x_j\}$)

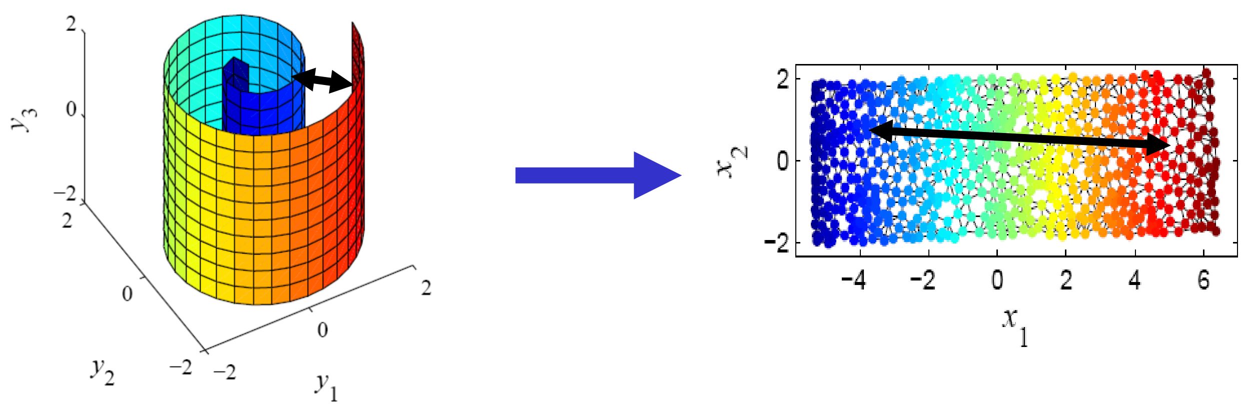
NLDR history

- Principal component analysis (PCA) 1901
- Classical metric multidimensional scaling (MDS) 1938
- Stress-based MDS 1952
- Nonmetric MDS 1962
- Sammon mapping 1969
- Self-organizing map 1982
- Principal curves 1984
- Auto-encoder (bottleneck FFN) 1991
- Curvilinear component analysis 1993
- Spectral methods
  - Kernel PCA 1996
  - Isomap 1998
  - Locally linear embedding 2000
  - Laplacian eigenmaps 2002
  - Maximum variance unfolding 2004
- Deep auto-encoder 2006
- Similarity-based embedding
  - (*t*-distributed) stochastic neighbor embedding 2008
  - Neignbor retrieval and visualization (NeRV) 2010
- etc.

- PCA (Principal Component Analysis), and MDS (classical MultiDimensional Scaling) preserve distances

$$\min_X \sum_{i<j} \left(\delta_{ij}^2 - d_{ij}^2\right)^2 \qquad \text{where} \quad \begin{aligned} \delta_{ij} &= \left\|y_i - y_j\right\| \text{ are the distances in the original space} \\ d_{ij} &= \left\|x_i - x_j\right\| \text{ are the distances in the projection space} \end{aligned}$$

- Nice and intuitive, but…
  - intuitively, local distances are much more interesting to preserve than distances between far away data
  - preserving distances does not allow to unfold

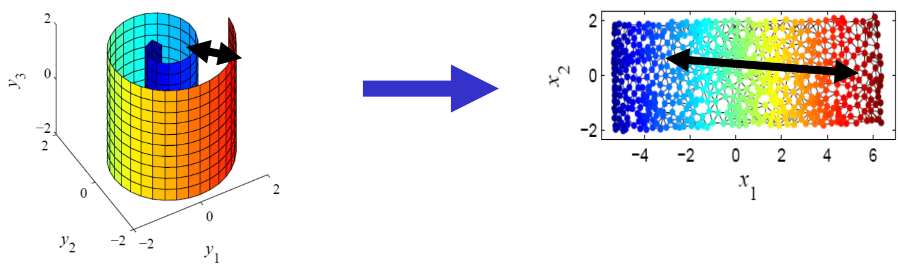


weighted distances, and similarities

$$E_{NLM} = \sum_{\substack{i=1 \\ i<j}}^{N} \frac{\left(\delta_{ij} - d_{ij}\right)^2}{\delta_{ij}} \qquad E_{CCA} = \sum_{\substack{i=1 \\ i<j}}^{N} \frac{\left(\delta_{ij} - d_{ij}\right)^2}{d_{ij}}$$

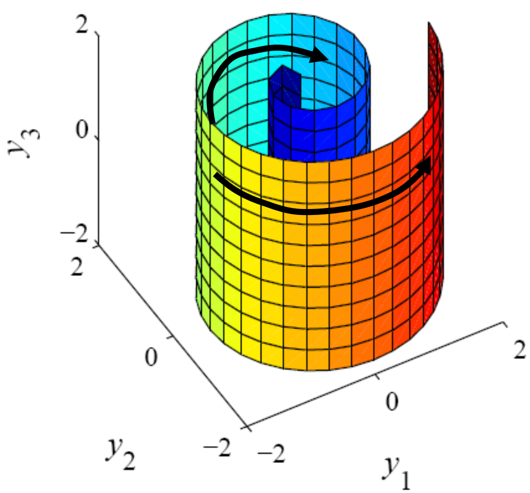
- Weighting by inverse distances concentrates of locality

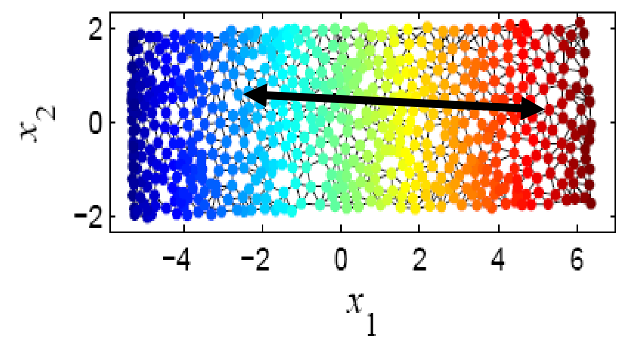
- There is a compromise between intrusions and extrusions
  - intrusions: linear projection, PCA”flattens” distributions
  - extrusions: close data will not be projected close



curvilinear distances

- Goal: to measure distances along the manifold
- Such distances are more easily preserved when unfolding





- Weighting and curvilinear distances may be combined !
(see for example Curvilinear Distance Analysis)

similarities

- Examples
  - t-distributed SNE (2008)
  - Neighbour retrieval and visualisation (NeRV, 2010)

- Ingredients
  - Softmax similarities:

$$\sigma_{ij} = \frac{\exp(-\delta_{ij}^2/(2\lambda_i^2))}{\sum_{k,k\neq i}\exp(-\delta_{ik}^2/(2\lambda_i^2))} \quad \text{and} \quad s_{ij} = \frac{(1+d_{ij}^2)^{-1}}{\sum_{k,l,k\neq l}(1+d_{kl}^2)^{-1}}$$

t-SNE

  - Allows to introduce hypotheses on the distribution of distances (remember the concentration of distances!)

  - Similarity preservation (sum of KL divergences):

$$E(\mathbf{X};\boldsymbol{\Xi},\boldsymbol{\Lambda}) = \sum_i E_i(\mathbf{X};\boldsymbol{\Xi},\lambda_i) = \sum_{i,j}\sigma_{ij}\log(\sigma_{ij}/s_{ij})$$

similarities

- t-SNE and similar methods: depend on a “scale” parameter that defines how far similarities are considered

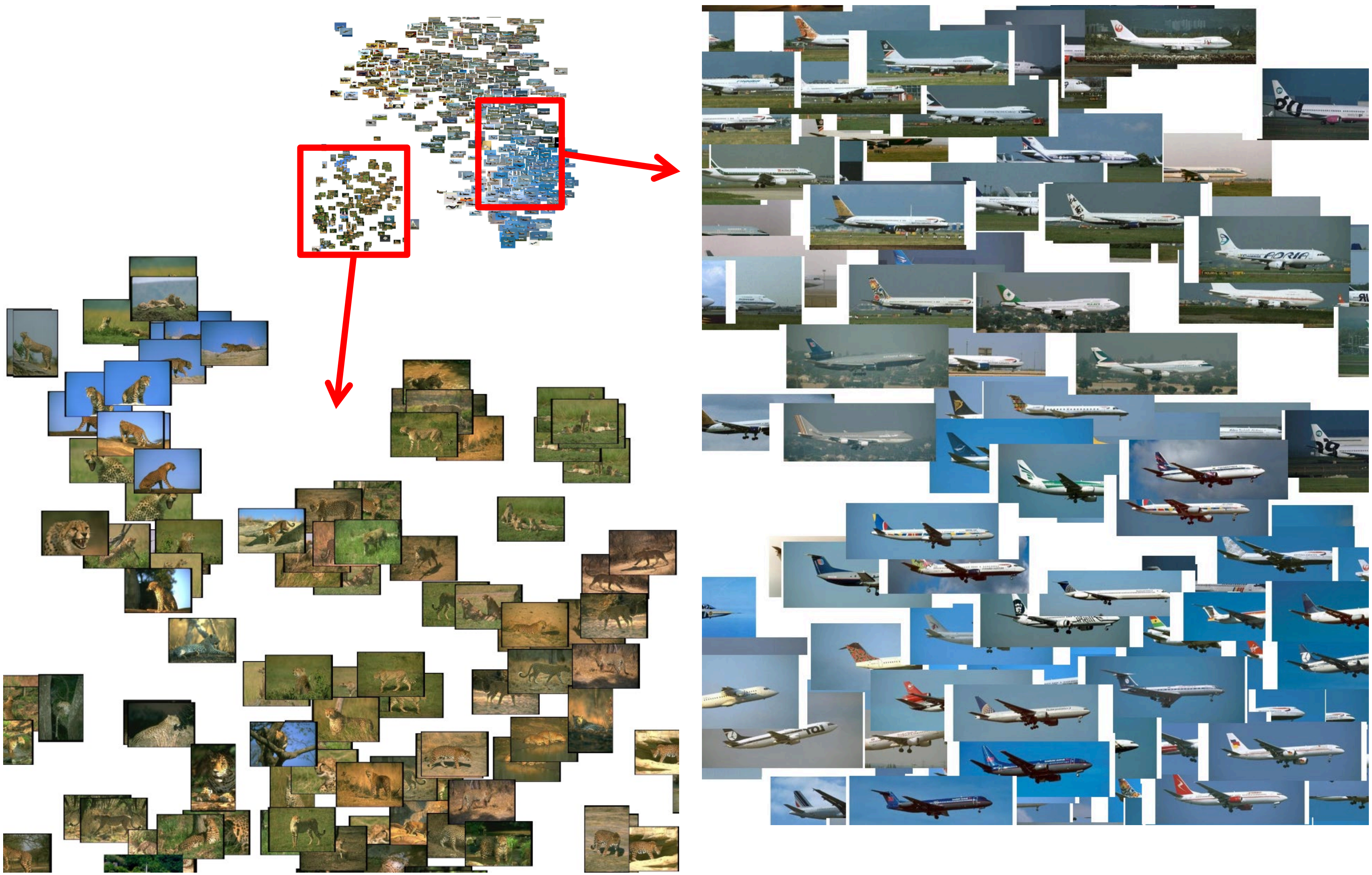
- possibility to integrate into “multi-scale t-SNE”: no more parameter!

Multiscale stochastic neighbor embedding: Towards parameter-free dimensionality reduction
John A. Lee et al., ESANN 2014

Image banks

L. Van der Maaten, 2012

Image banks



About the evaluation

- Feature selection and dimensionality reduction are unsupervised processes
  → evaluation not so easy

- Feature selection and dimensionality reduction
  - may be evaluated by the quality of the model (on the selected features)
  - but what about feature discovery, visualization,…

- Nonlinear dimensionality reduction
  - evaluation by neighbourhood preservation

  Quality assessment of dimensionality reduction: Rank-based criteria

  John A. Lee & Michel Verleysen, Neurocomputing, Volume 72, Issues 7–9, March 2009, Pages 1431-1443

Outline

- High-dimensional data analysis
- The curse of dimensionality
- Feature selection
- Nonlinear dimensionality reduction
- Other small *n*, large *p* issues in machine learning
- Conclusion

Other small n large p issues

- few data are labelled, lots of data are unlabelled
  ex.: medical data, images, etc. costly to label
  → semi-supervised learning

- few good data, many similar but not identical data
  ex.: many "old" data, few recent ones, non-stationary process
  → transfer learning

- missing data
  ex.: out of order sensors, successive medical tests,…
  → data completion or (better) models robust to outliers

- still too few data
  → data generation, oversampling, GAN networks,…

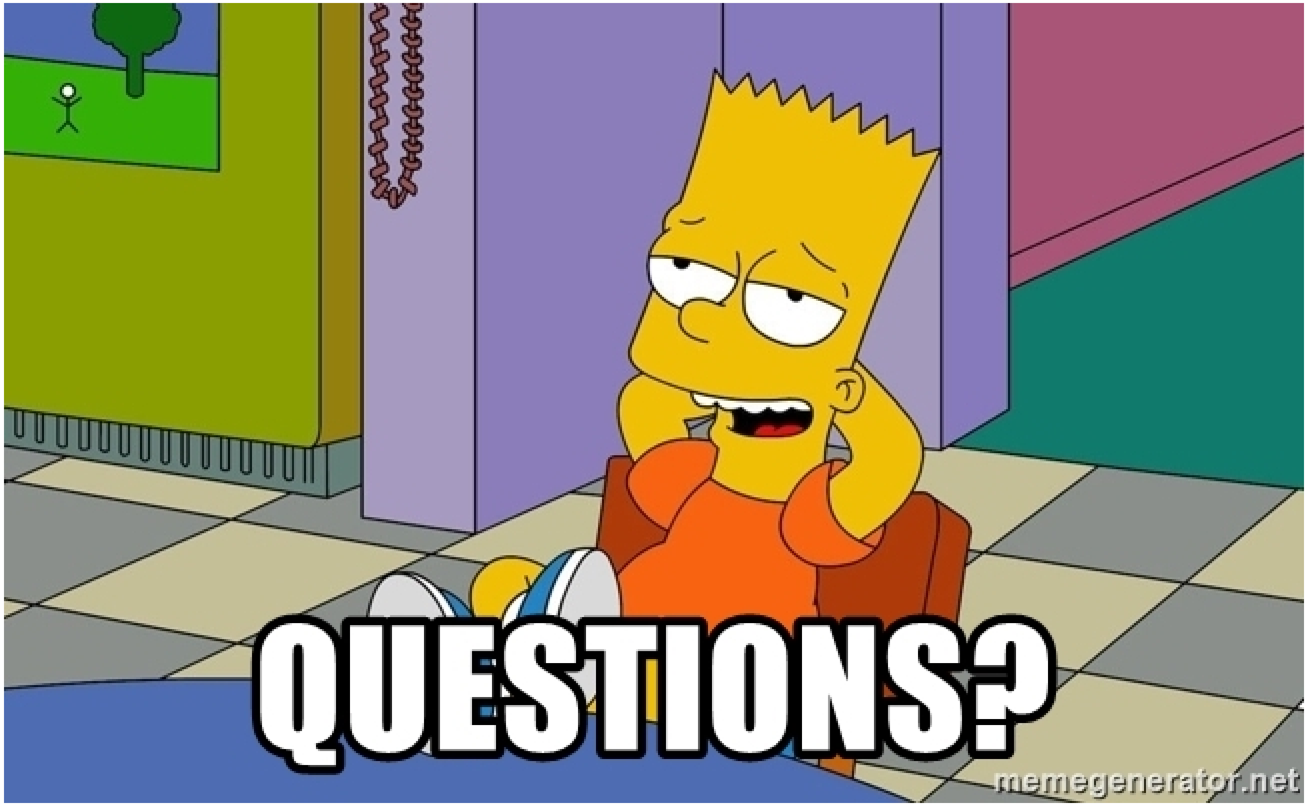
Take-home messages

- Machine learning is not only about big data and deep learning
- Many real-world domains do not provide big data, still they are interesting
- Model-based learning is a fascinating and still very open field (of research and application)
- Do not underestimate the need to *understand* data (in addition to building good models)

Thanks to:

This work would not have been possible without the help of my colleague John A. Lee,

and numerous PhD students and post-docs


QUESTIONS?
memegenerator.net