

UVA CS 4774: Machine Learning

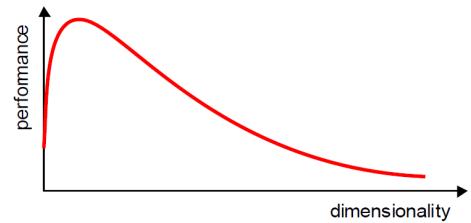
Lecture 14: Dimension Reduction

Dr. Yanjun Qi

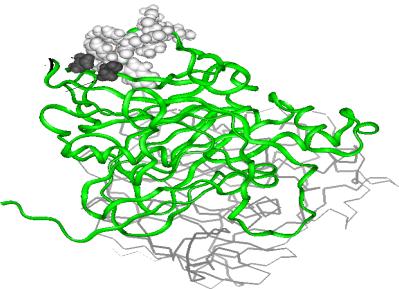
University of Virginia
Department of Computer Science

Curse of Dimensionality

- Increasing the number of features will not always improve classification accuracy.
- In practice, the inclusion of more features might actually lead to **worse** performance.
- The number of training examples required increases **exponentially** with dimensionality p



e.g., QSAR: Drug Screening

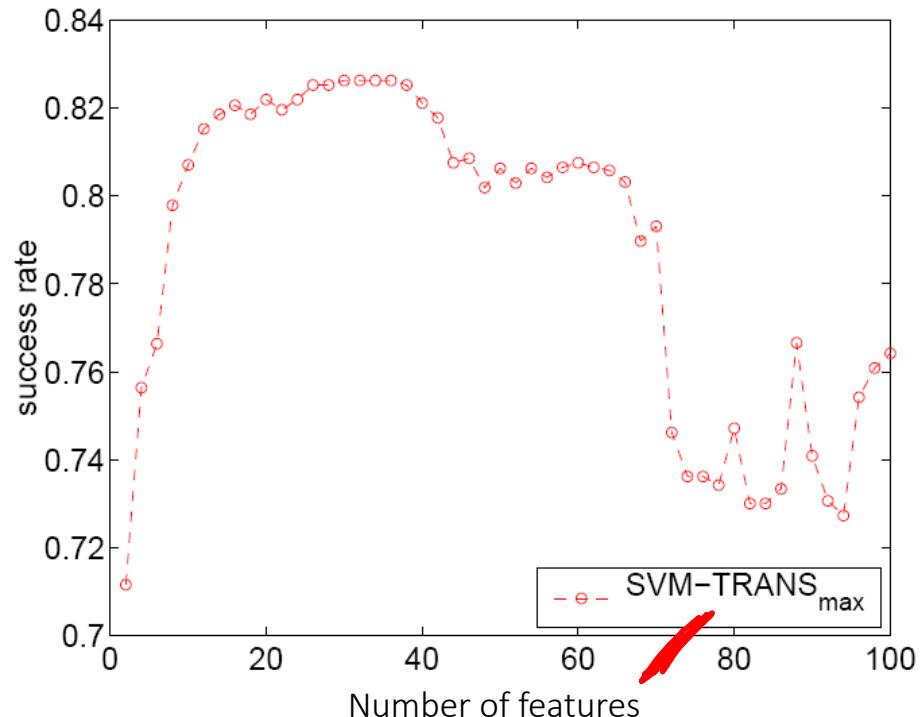


Binding to Thrombin

(DuPont Pharmaceuticals)

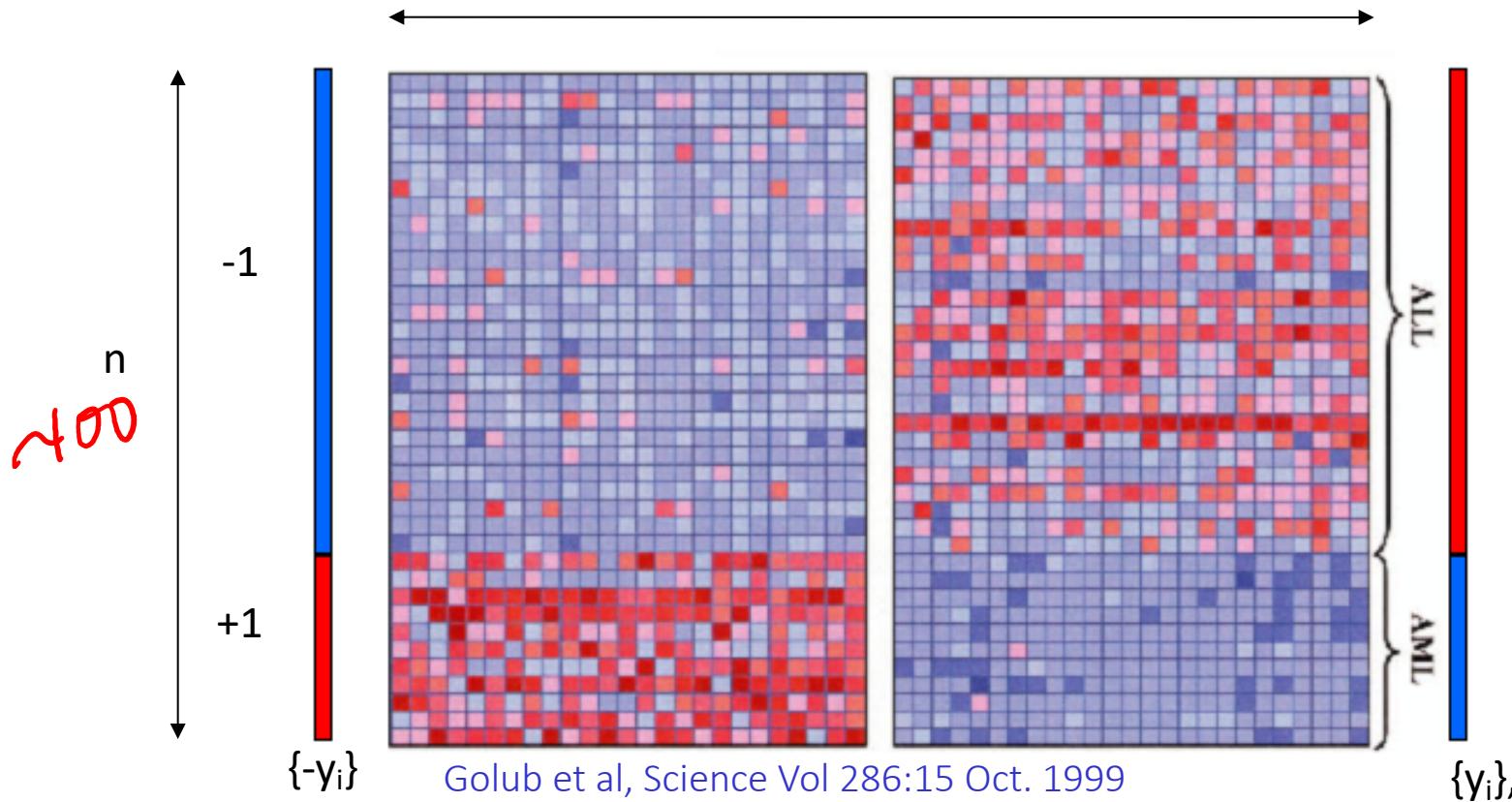
[2543 compounds tested] for their ability to bind to a target site on thrombin, a key receptor in blood clotting; 192 “active” (bind well); the rest “inactive”. Training set (1909 compounds) more depleted in active compounds.

[139,351 binary features], which describe three-dimensional properties of the molecule.

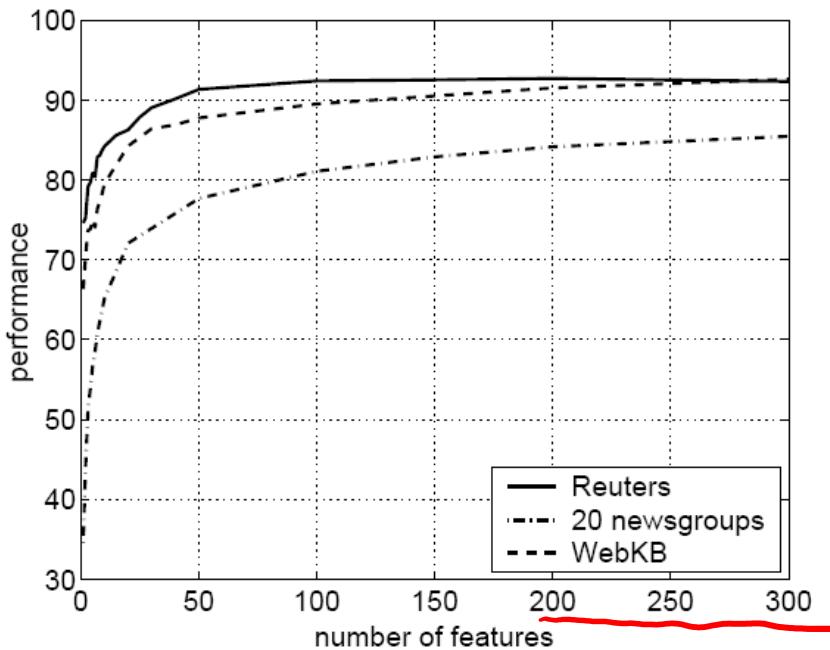


e.g., Leukemia Diagnosis

$p' \sim 20k$



e.g., Text Categorization with many BOW features



Reuters: 21578 news wire, 114 semantic categories.

20 newsgroups: [19997] articles, 20 categories.
n

WebKB: 8282 web pages, 7 categories.

Bag-of-words: >100,000 features.

[↑P)

Bekkerman et al,
JMLR, 2003

e.g., Movie Reviews and Revenues: An Experiment in Text Regression, Proceedings of HLT '10 (1.7k n / >3k features)

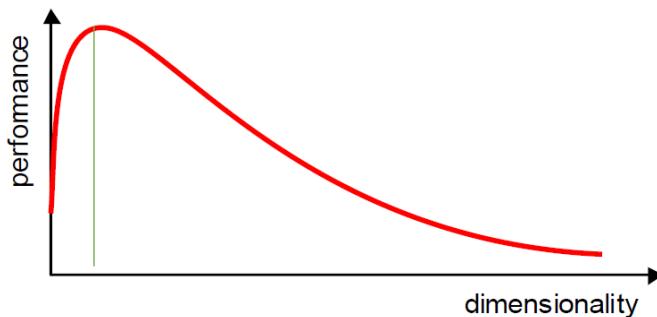
IV. Features

I	Lexical n-grams (1,2,3)
II	Part-of-speech n-grams (1,2,3)
III	Dependency relations (nsubj,advmod,...)
Meta	U.S. origin, running time, budget (log), # of opening screens, genre, MPAA rating, holiday release (summer, Christmas, Memorial day,...), star power (Oscar winners, high-grossing actors)

e.g. counts
of a ngram in
the text

Dimensionality Reduction

- What is the objective?
 - Choose an optimum set of features of lower dimensionality to **improve** classification accuracy.



Dimension Reduction → Simpler models

- Because:
 - Simpler to use (lower computational complexity)
 - Easier to train (needs less examples)
 - Less sensitive to noise
 - Easier to explain (more interpretable)
 - Generalizes better (lower variance)

Today: Dimensionality Reduction (Two Ways)

Feature extraction: finds a set of **new** features (i.e., through some mapping $f()$) from the **existing** features.

Feature selection: chooses a subset of the **original** features.



The mapping $f()$ could be linear or non-linear

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} \xrightarrow{f()} \mathbf{h} = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_K \end{bmatrix}$$

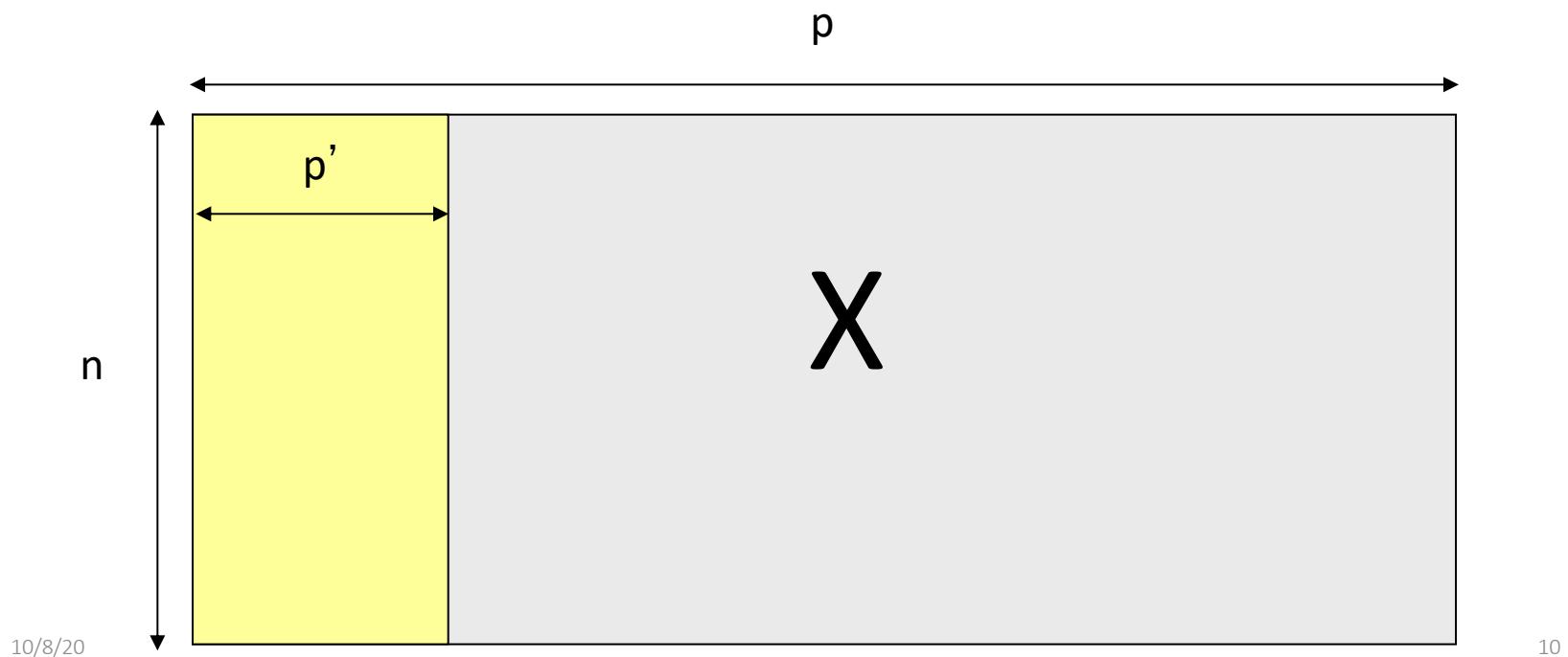
$K \ll N$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} \longrightarrow \mathbf{x}' = \begin{bmatrix} x_{k1} \\ x_{k2} \\ \vdots \\ x_{kK} \end{bmatrix}$$

$K \ll N$

Feature Selection

- Select the most relevant ones to build **better, faster, and easier to understand** learning models.

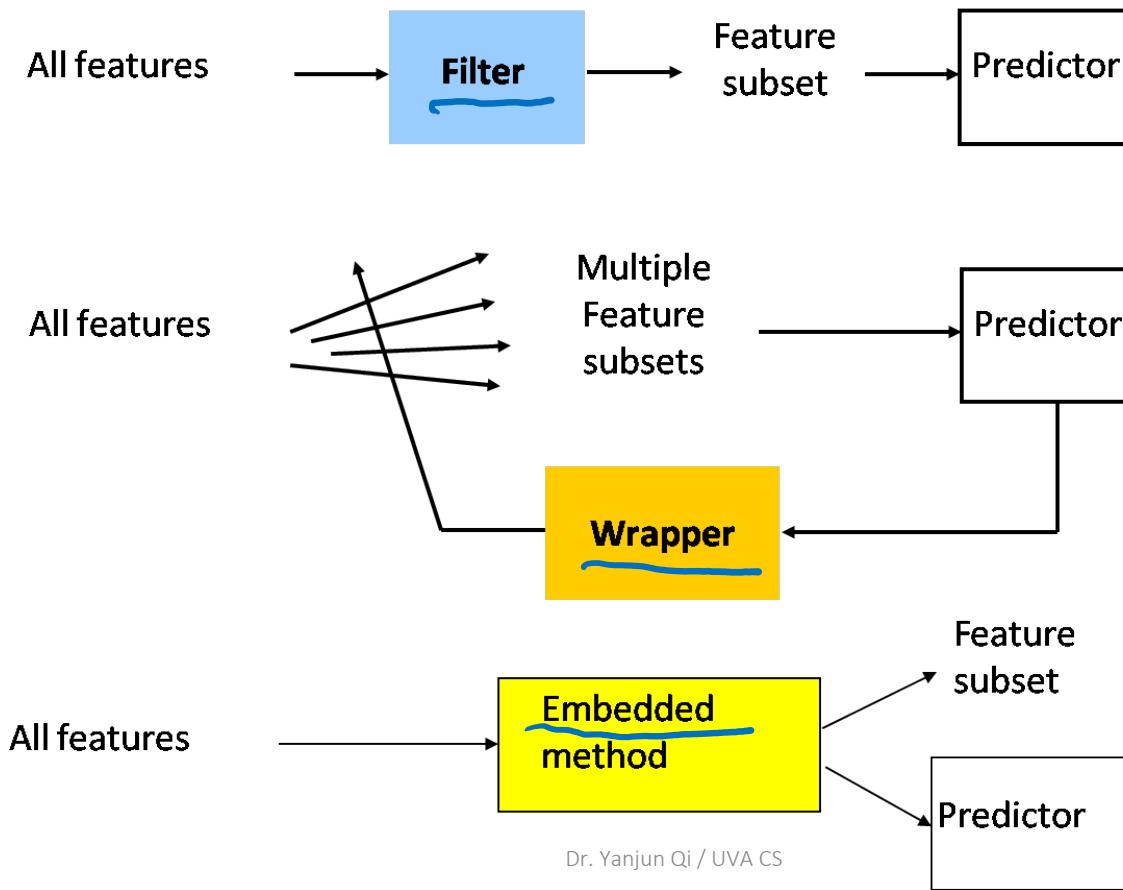


Summary: Feature Selection

- Filtering approach:
ranks features or feature subsets **independently of** the predictor.
 - ...using **univariate** methods: consider **one** variable at a time
 - ...using **multivariate** methods: consider **more than one** variables at a time
- Wrapper approach:
uses a **predictor to assess (many)** features or feature subsets.
- Embedding approach:
uses a **predictor to build** a (single) model with a subset of features that are internally selected.

Summary: filters vs. wrappers vs. embedding

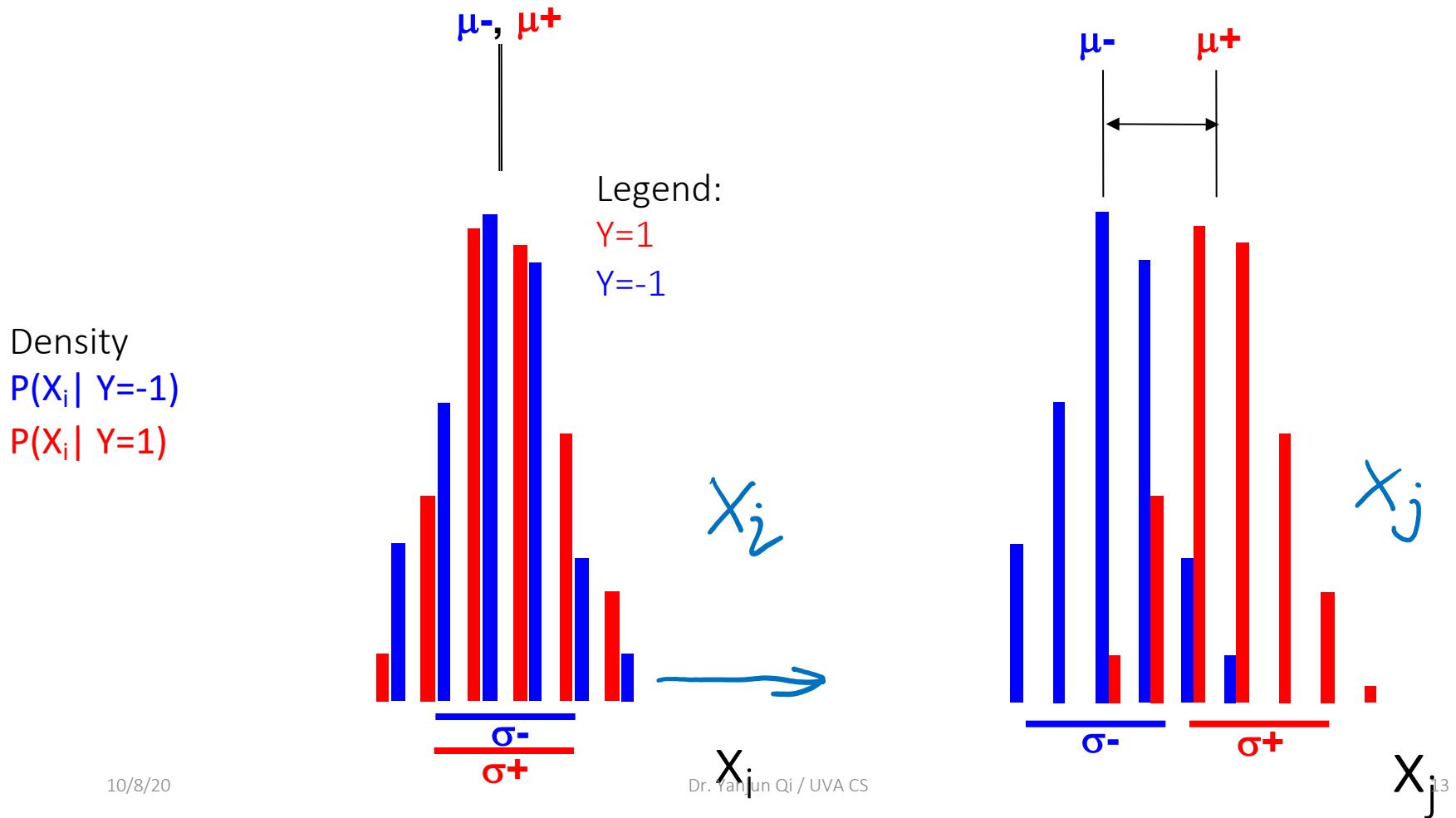
- Main goal: rank subsets of useful features



(I) Filtering: univariate filtering

e.g. T-test

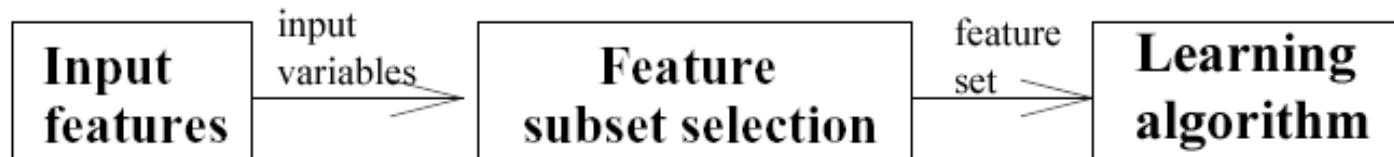
- Goal: determine the relevance of a given single feature for two classes of samples.



(I) Filtering : multi-variate: Feature Subset Selection

Filter Methods

- Select subsets of variables as a pre-processing step, independently of the used classifier!!



- E.g. Group correlation
- E.g. Information theoretic filtering methods such as Markov blanket

(I) Filtering : Summary

Filter Methods

- usually fast
- provide generic selection of features, not tuned by given learner (universal)
- this is also often criticised (feature set not optimized for used learner)
- Often used as a preprocessing step for other methods

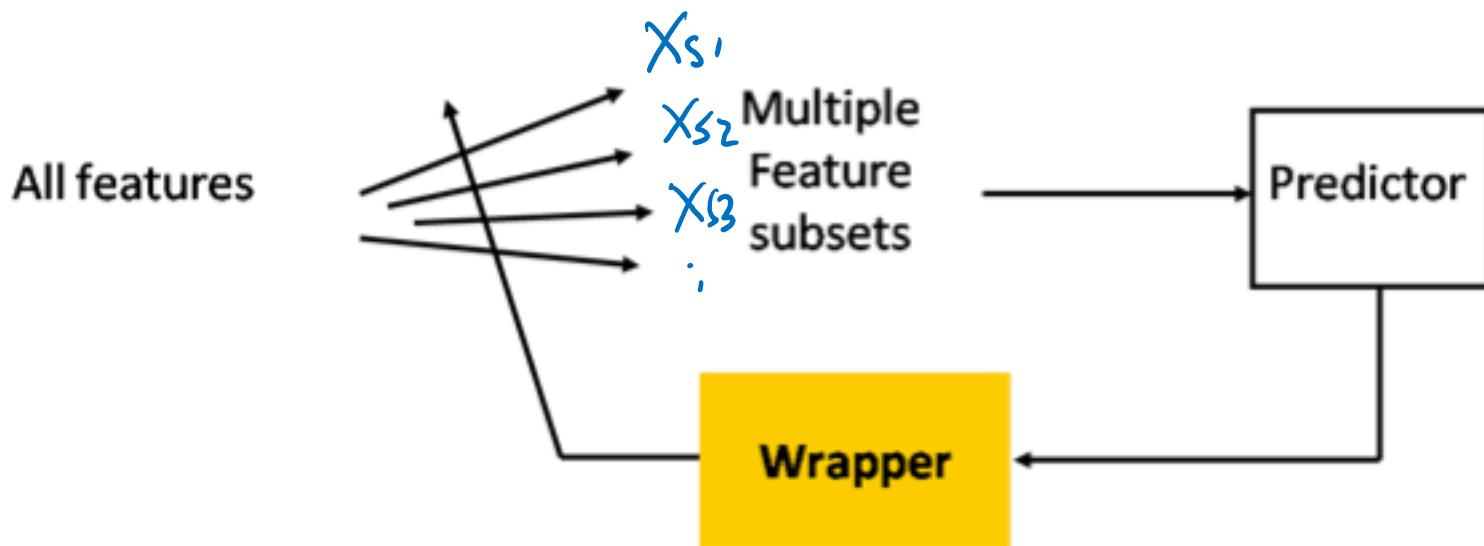
(I) Filtering : (many choices)

Method	X	Y	Comments
Name	Formula	B M C B M C	
Bayesian accuracy	Eq. 3.1	+ s + s	Theoretically the golden standard, rescaled Bayesian relevance Eq. 3.2.
Balanced accuracy	Eq. 3.4	+ s + s	Average of sensitivity and specificity; used for unbalanced dataset, same as AUC for binary targets.
Bi-normal separation	Eq. 3.5	+ s + s	Used in information retrieval.
F-measure	✓	Eq. 3.7 + s + s	Harmonic of recall and precision, popular in information retrieval.
Odds ratio	✓	Eq. 3.6 + s + s	Popular in information retrieval.
Means separation	Eq. 3.10	+ i + +	Based on two class means, related to Fisher's criterion.
T-statistics	Eq. 3.11	+ i + +	Based also on the means separation.
Pearson correlation	✓	Eq. 3.9 + i + + i +	Linear correlation, significance test Eq. 3.12, or a permutation test.
Group correlation	✓	Eq. 3.13 + i + + i +	Pearson's coefficient for subset of features.
χ^2	✓	Eq. 3.8 + s + s	Results depend on the number of samples m .
Relief		Eq. 3.15 + s + + s +	Family of methods, the formula is for a simplified version ReliefX, captures local correlations and feature interactions.
Separability Split Value	Eq. 3.41	+ s + + s	Decision tree index.
Kolmogorov distance	Eq. 3.16	+ s + + s +	Difference between joint and product probabilities.
Bayesian measure	Eq. 3.16	+ s + + s +	Same as Vajda entropy Eq. 3.23 and Gini Eq. 3.39.
Kullback-Leibler divergence	Eq. 3.20	+ s + + s +	Equivalent to mutual information.
Jeffreys-Matusita distance	Eq. 3.22	+ s + + s +	Rarely used but worth trying.
Value Difference Metric	Eq. 3.22	+ s + s	Used for symbolic data in similarity-based methods, and symbolic feature-feature correlations.
Mutual Information	✓	Eq. 3.29 + s + + s +	Equivalent to information gain Eq. 3.30.
Information Gain Ratio	✓	Eq. 3.32 + s + + s +	Information gain divided by feature entropy, stable evaluation.
Symmetrical Uncertainty		Eq. 3.35 + s + + s +	Low bias for multivalued features.
J-measure		Eq. 3.36 + s + + s +	Measures information provided by a logical rule.
Weight of evidence	10/8/20	Eq. 3.37 + s + + s +	So far rarely used.
MDL		Eq. 3.38 + s + s	Dr. Yanjun Qi / UVA CS Low bias for multivalued features.

(2) Wrapper

- Wrapper approach:
uses a **predictor** to assess (many) features or feature subsets.

Wrapper Methods



(2) Wrapper : Feature Subset Selection

Wrapper Methods

- Learner is considered a black-box
- Interface of the black-box is used to score subsets of variables according to the predictive power of the learner when using the subsets.
- Results vary for different learners

(b). Search: even more search strategies for selecting feature subset

$P \rightarrow 2^P$ feature Subsets

- **Forward selection or backward elimination.**
- **Beam search:** keep k best path at each step.
- **GSFS:** generalized sequential forward selection – when (n-k) features are left try all subsets of g features. More trainings at each step, but fewer steps.
- **PTA(l,r):** plus l , take away r – at each step, run SFS l times then SBS r times.
- **Floating search:** One step of SFS (resp. SBS), then SBS (resp. SFS) as long as we find better subsets than those of the same size obtained so far.

(3) Embedded

- Embedding approach:
uses a **predictor to build** a (single) model
with a subset of features that are internally
selected.

lasso

elastiNet

In practice...

- No method is universally better:
 - wide variety of types of variables, data distributions, learning machines, and objectives.
- Feature selection is not always necessary to achieve good performance.

Today: Dimensionality Reduction (Two Ways)

Feature extraction: finds a set of **new** features (i.e., through some mapping $f()$) from the **existing** features.



Feature selection: chooses a subset of the **original** features.

The mapping $f()$ could be linear or non-linear

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} \xrightarrow{f()} \mathbf{h} = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_K \end{bmatrix}$$

$K \ll N$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} \longrightarrow \mathbf{x}' = \begin{bmatrix} x_{k1} \\ x_{k2} \\ \vdots \\ x_{kK} \end{bmatrix}$$

$K \ll N$

Feature Extraction

- Linear combinations are particularly attractive because they are simpler to compute and analytically tractable.

$\rightarrow p \times K$

- Given $x \in R^p$, find an $N \times K$ matrix U such that:

$$y = U^T x \in R^K \text{ where } K < P$$

~~R^P~~
 $R^P \xrightarrow{f(x)} R^K$

This is a projection from the N-dimensional space to a K-dimensional space.

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} \xrightarrow{U^T} h = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_K \end{bmatrix}$$

$f(x) = U^T x$
 $P \times 1$

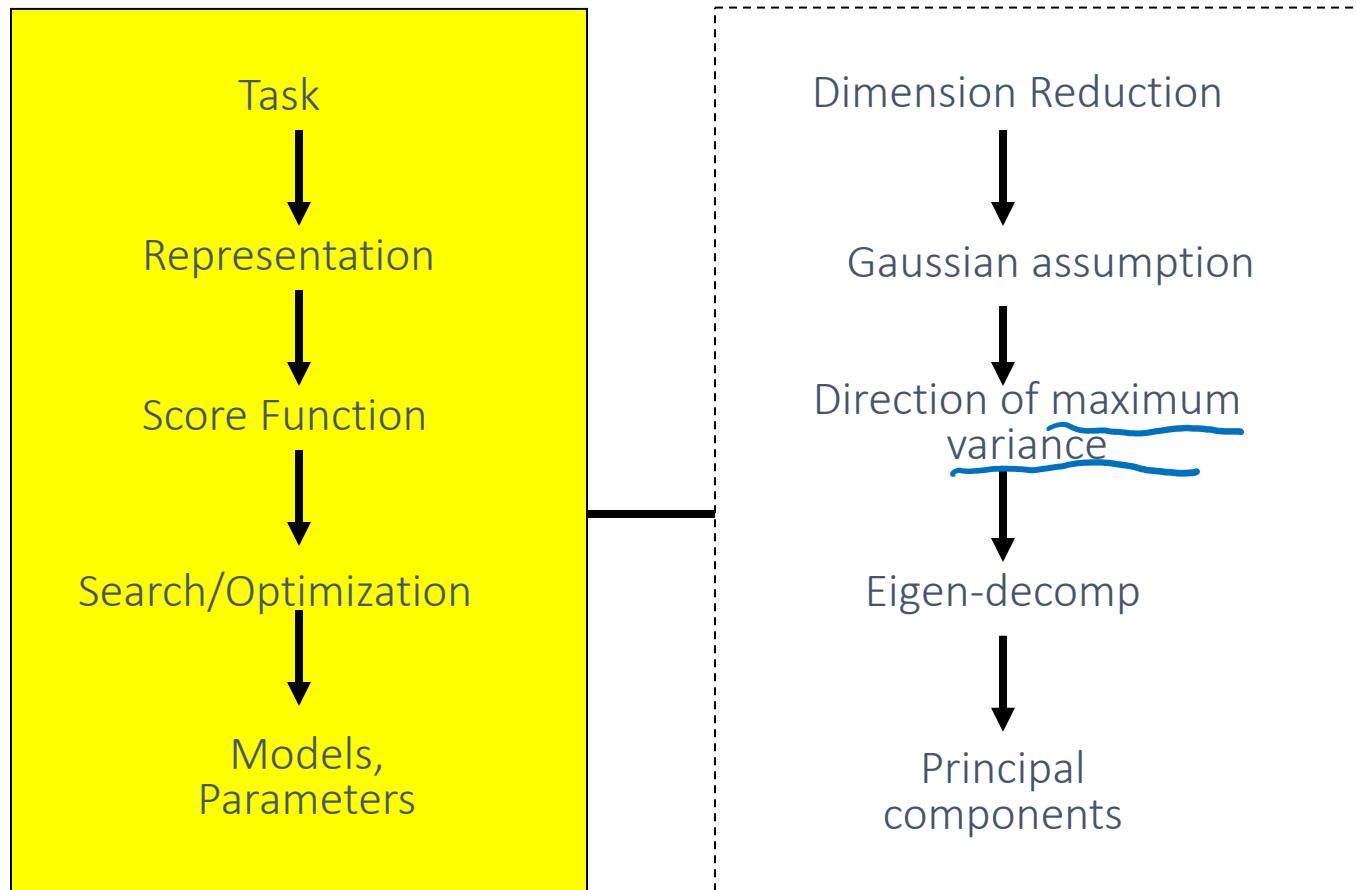
Feature Extraction (cont'd)

- From a mathematical point of view, finding an **optimum** mapping $\mathbf{h} = f(\mathbf{x})$ is equivalent to optimizing an **objective** function.
- Different methods use different objective functions, e.g.,
 - **Information Loss**: The goal is to represent the data as accurately as possible (i.e., no loss of information) in the lower-dimensional space.
 - **Discriminatory Information**: The goal is to enhance the class-discriminatory information in the lower-dimensional space.

Feature Extraction (cont'd)

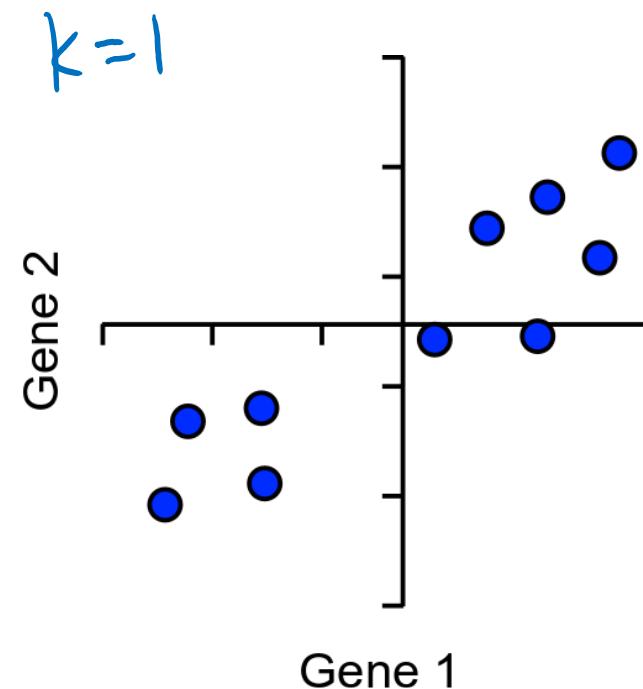
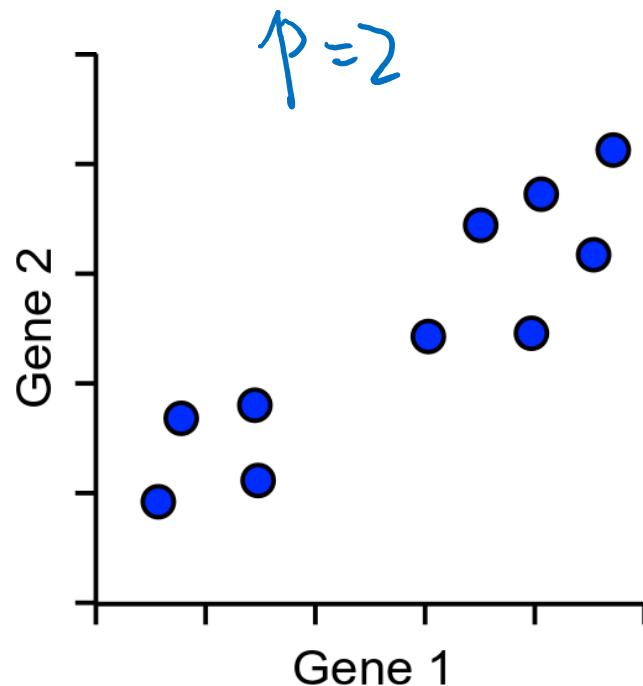
- Commonly used **linear** feature extraction methods:
 - Principal Components Analysis (PCA): Seeks a projection that **preserves** as much **information** in the data as possible.
 - Linear Discriminant Analysis (LDA): Seeks a projection that **best discriminates** the data.
- More methods:
 - Retaining interesting directions (**Projection Pursuit**),
 - Making features as independent as possible (**Independent Component Analysis or ICA**),
 - Embedding to lower dimensional manifolds (**Isomap, Locally Linear Embedding or LLE**).

Principal Component Analysis



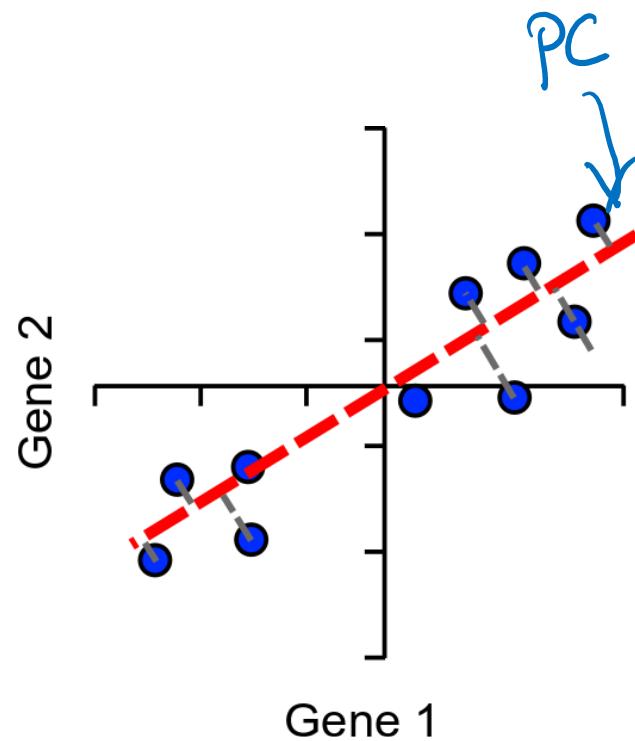
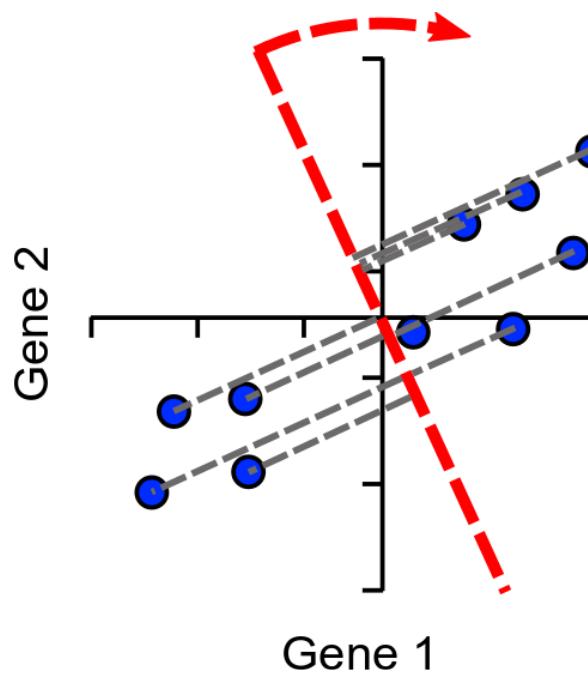
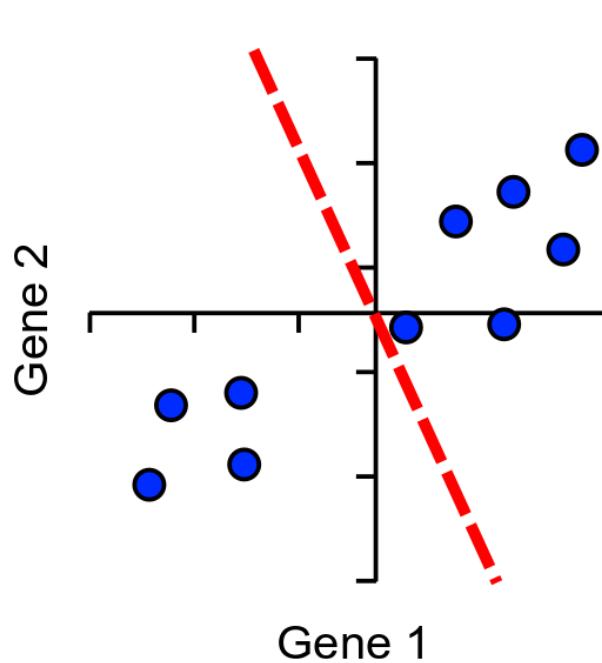
How does PCA work?

- Principal Components Analysis (PCA): approximating a high-dimensional data set with a lower-dimensional linear subspace



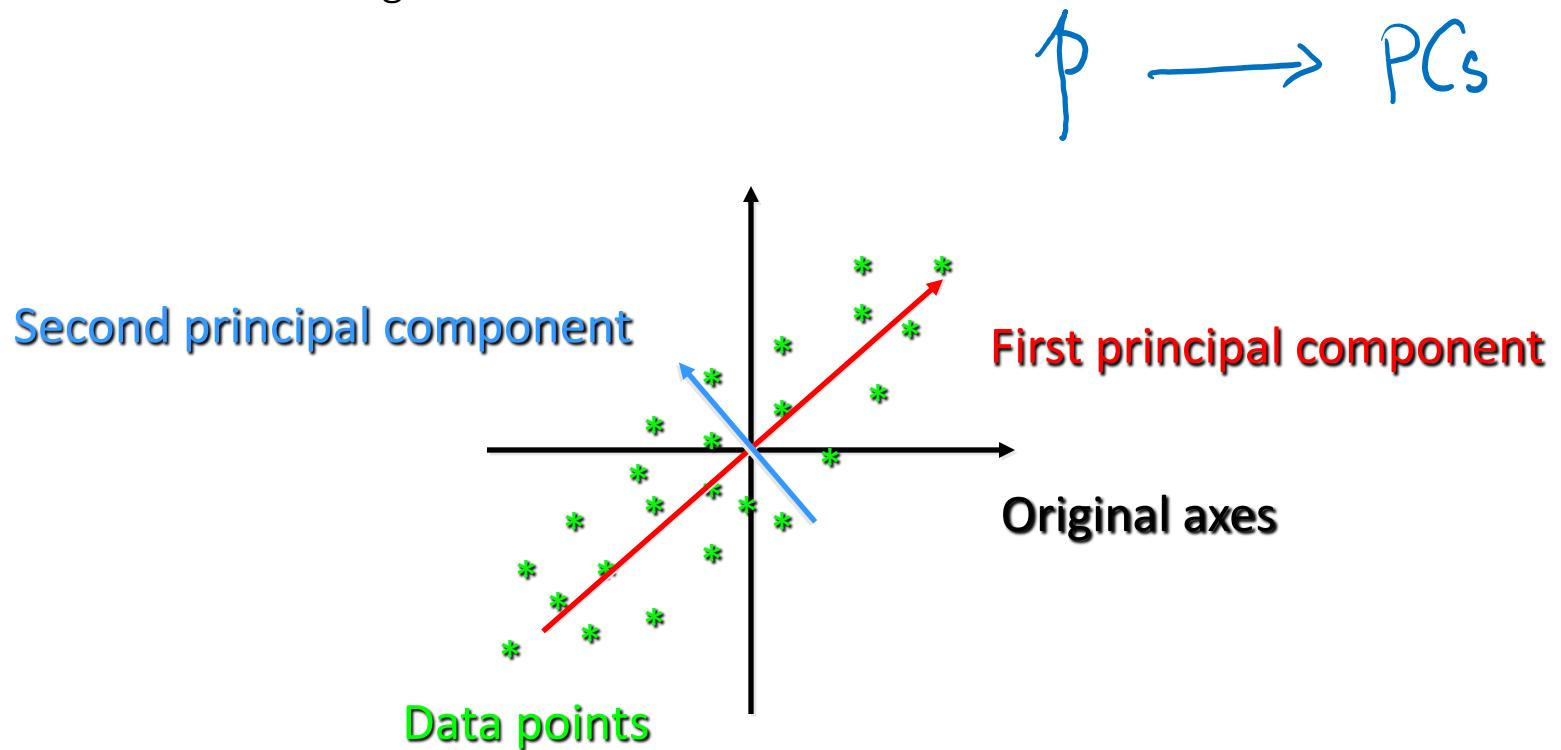
How does PCA work?

- Find line of best fit, passing through the origin

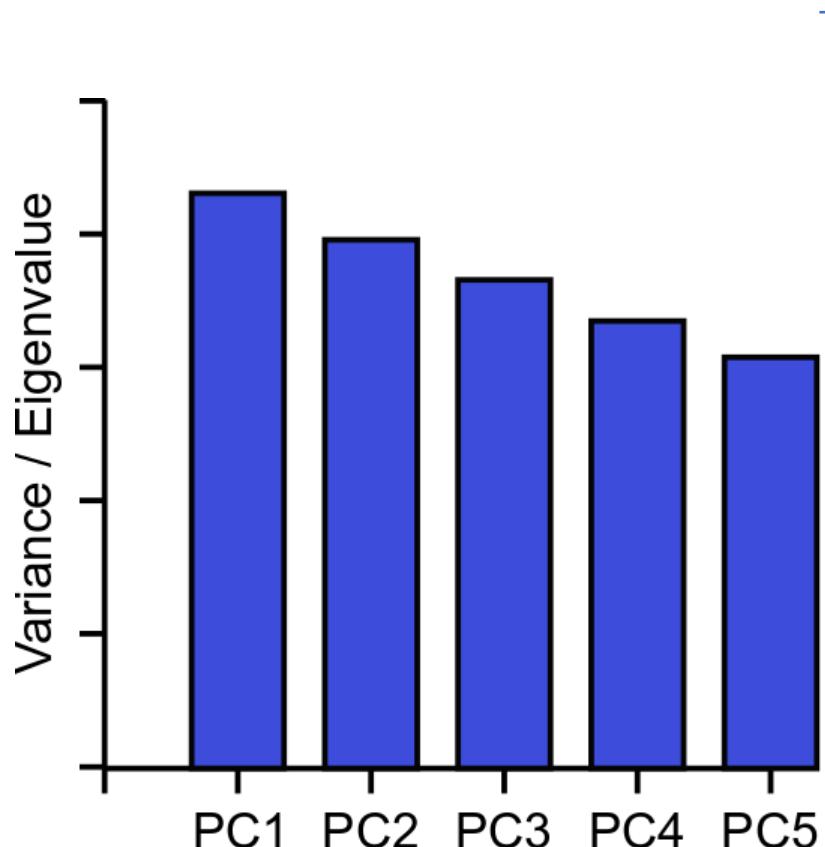
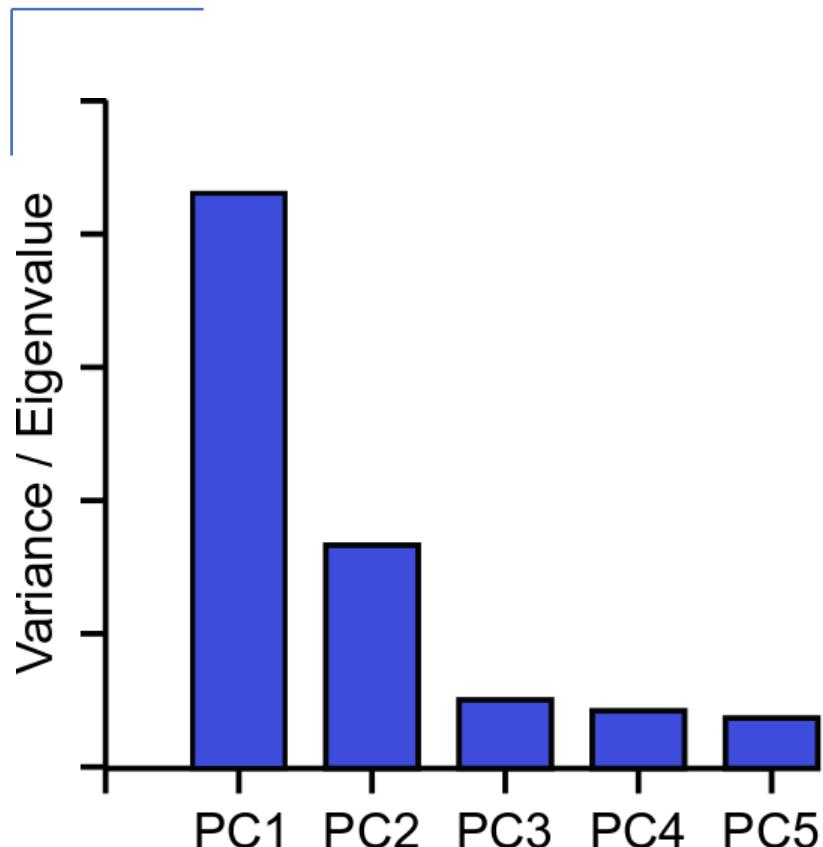


How does PCA work? Explaining Variance

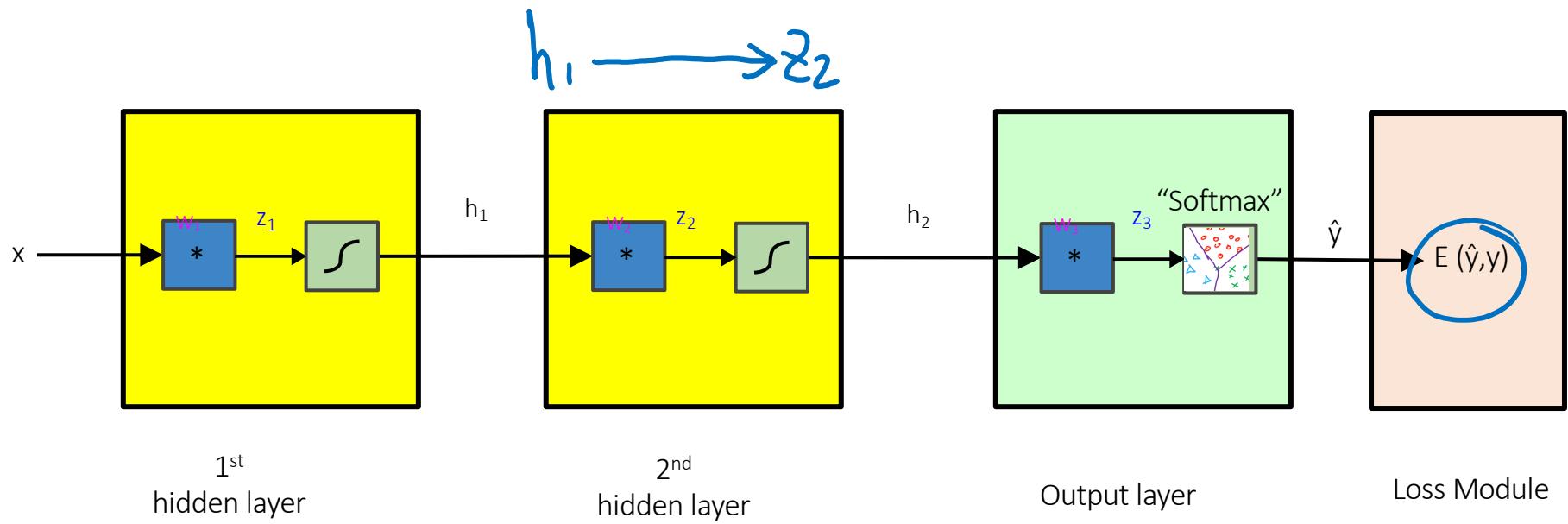
- Each PC always explains some proportion of the total variance in the data. Between them they explain everything
 - PC1 always explains the most
 - PC2 is the next highest etc. etc.



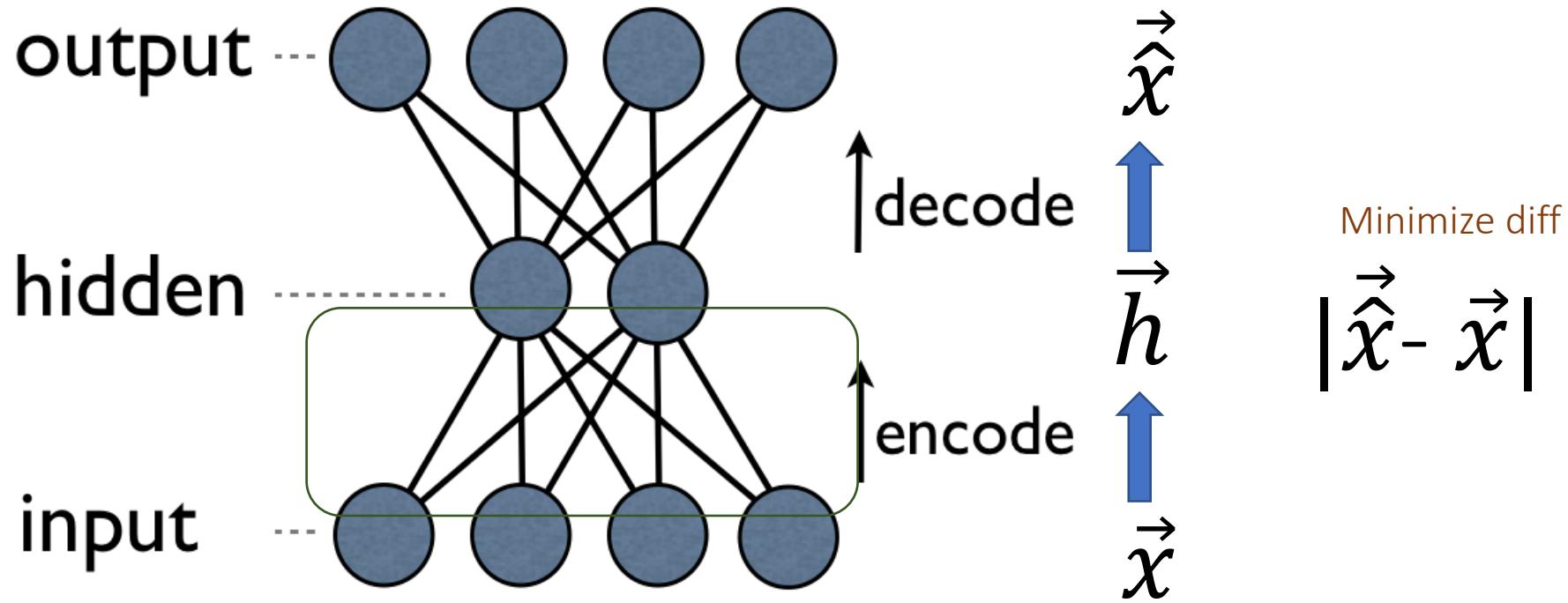
Explaining Variance – Scree Plots



Recap: “Block View”



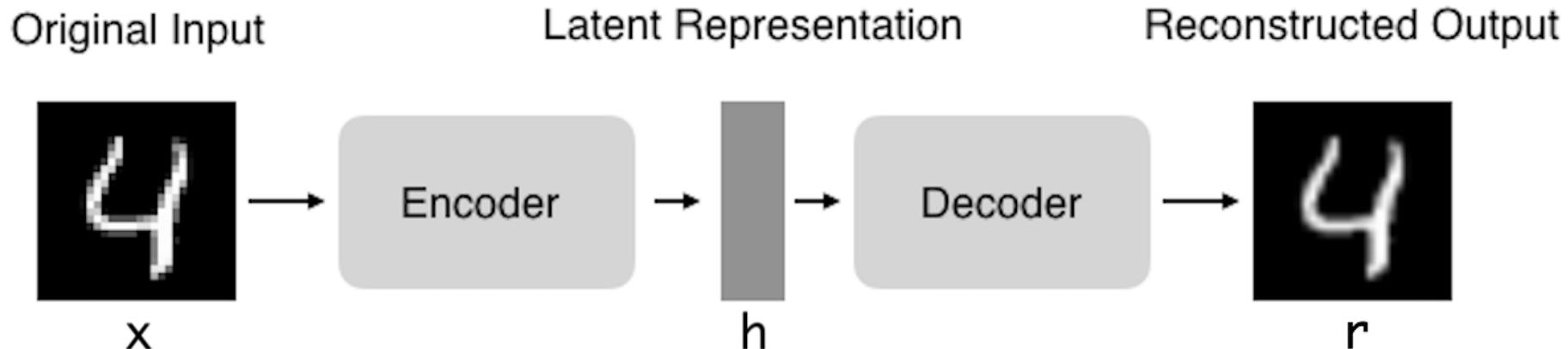
an auto-encoder-decoder is trained to reproduce the input



[Reconstruction Loss] force the ‘hidden layer’ units to become good / reliable feature detectors

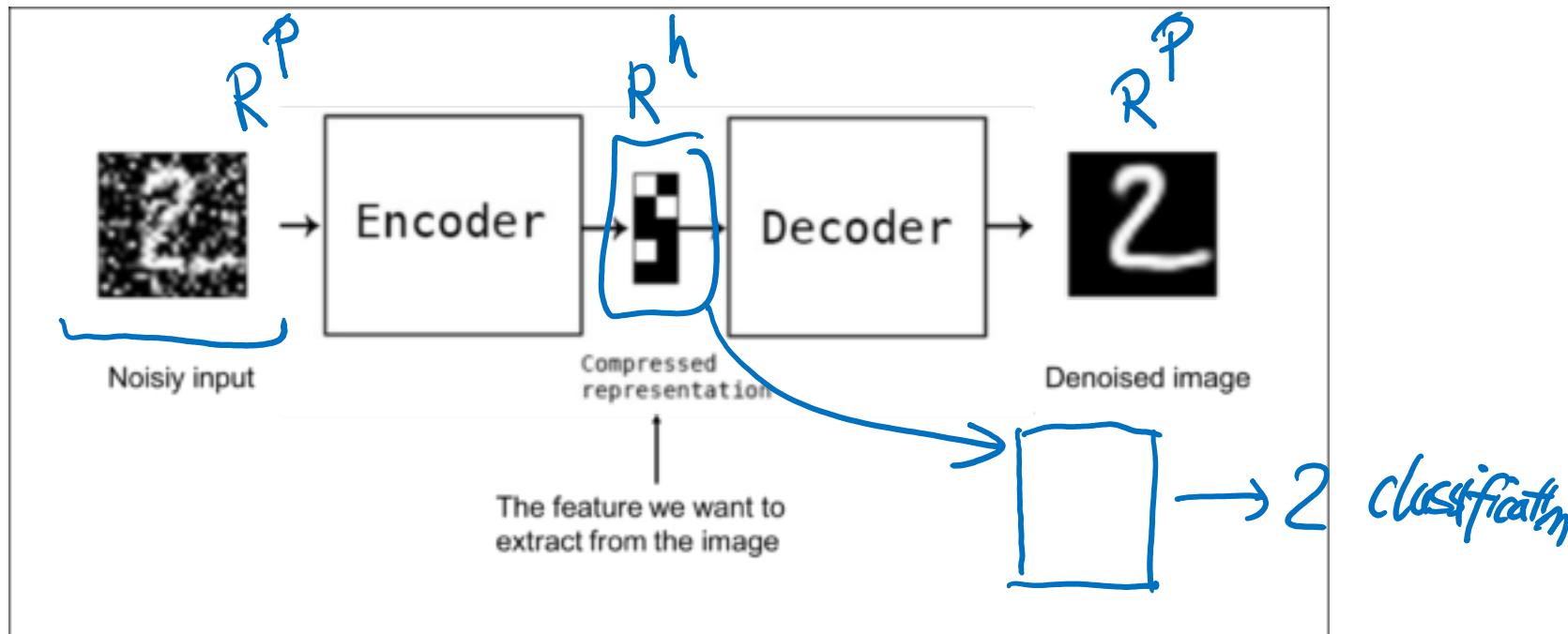
Autoencoders: structure

- Encoder: compress input into a latent-space of usually smaller dimension. $h = f(x)$
- Decoder: reconstruct input from the latent space. $r = g(f(x))$ with r as close to x as possible



Autoencoders: many variations

- Denoising: input clean image + noise and train to reproduce the clean image.
- Neural network autoencoders
 - Can learn nonlinear dependencies
 - Can use convolutional layers
 - Can use transfer learning



Today Recap: Dimensionality Reduction (Two Ways)

Feature extraction: finds a set of **new** features (i.e., through some mapping $f()$) from the **existing** features.

Feature selection: chooses a subset of the **original** features.

The mapping $f()$
could be linear or
non-linear

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} \xrightarrow{f()} \mathbf{h} = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_K \end{bmatrix}$$

$K \ll N$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} \longrightarrow \mathbf{x}' = \begin{bmatrix} x_{k1} \\ x_{k2} \\ \vdots \\ x_{kK} \end{bmatrix}$$

$K \ll N$

Thank You

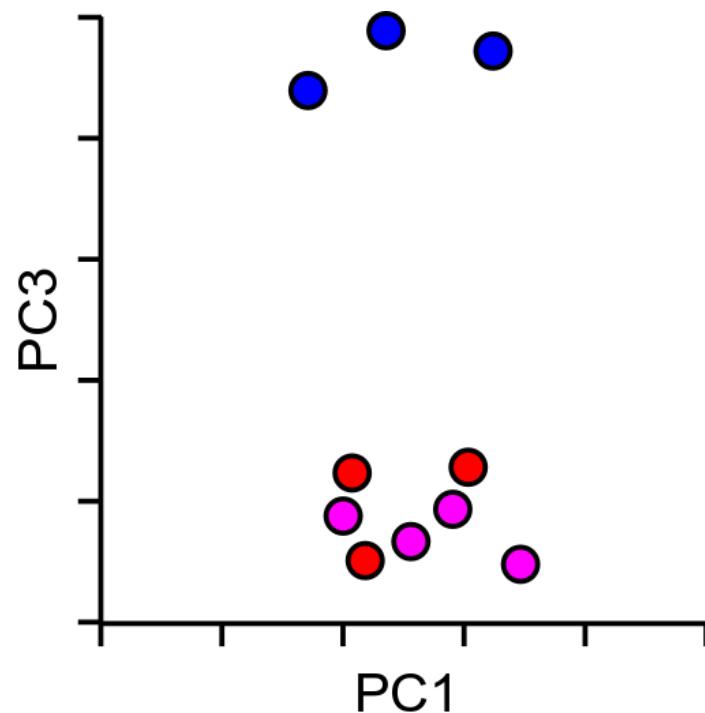
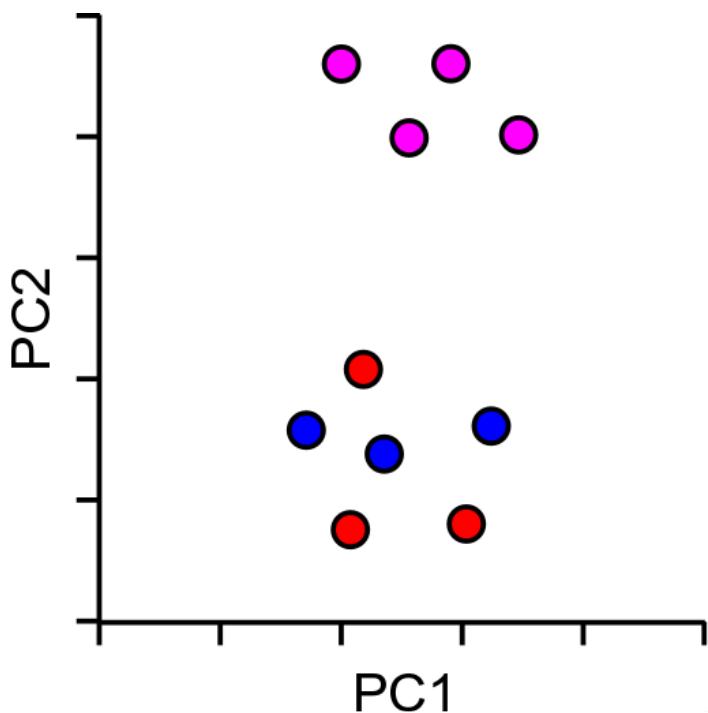


References

- Hastie, Trevor, et al. *The elements of statistical learning*. Vol. 2. No. 1. New York: Springer, 2009.
- Dr. S. Narasimhan's PCA lectures
- Prof. Derek Hoiem's eigenface lecture

So PCA is great then?

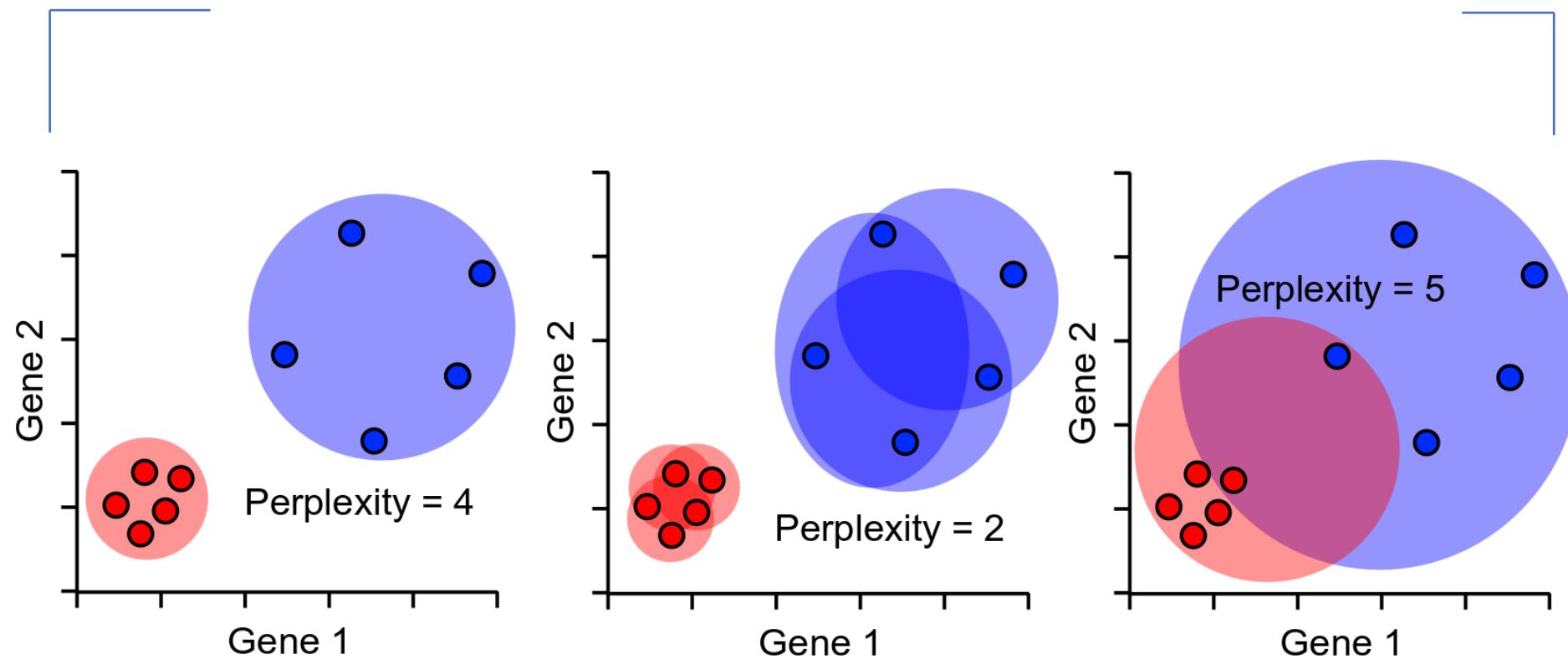
- Kind of...



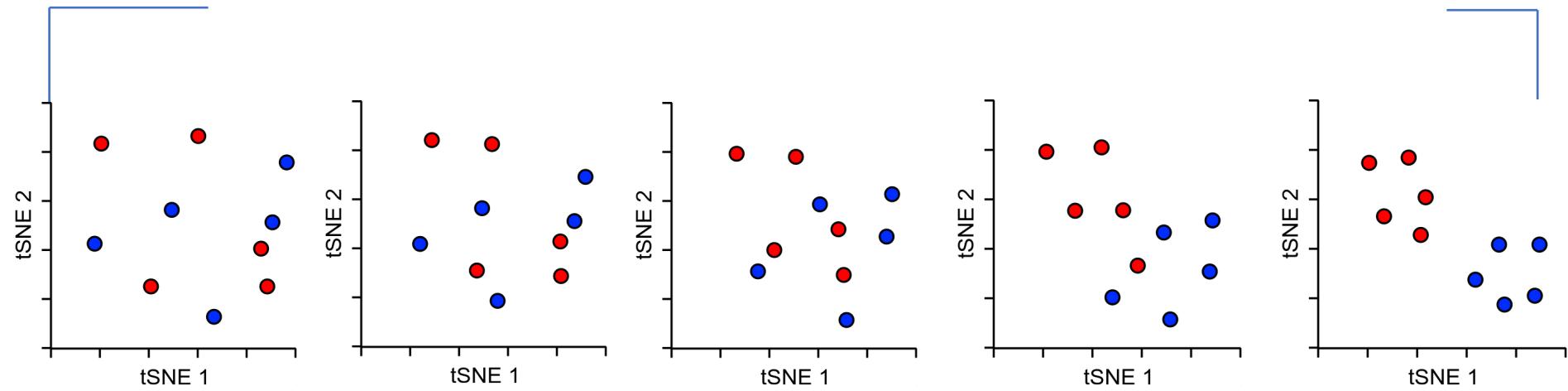
tSNE to the rescue...

- T-Distributed Stochastic Neighbour Embedding
- Aims to solve the problems of PCA
 - Non-linear scaling to represent changes at different levels
 - Optimal separation in 2-dimensions

Perplexity Robustness



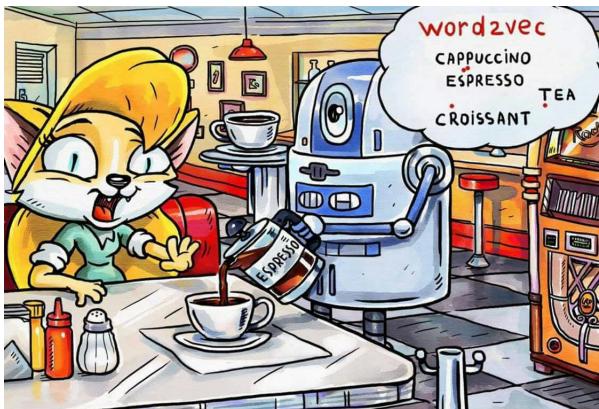
tSNE Projection



- X and Y don't mean anything (unlike PCA)
- Distance doesn't mean anything (unlike PCA)
- Close proximity is highly informative
- Distant proximity isn't very interesting
- Can't rationalise distances, or add in more data

Word2vec

- Input: large corpus of text
- Embed words into a high-dim space
 - words with common contexts in the corpus are close in the space



- Espresso? But I ordered a cappuccino!
- Don't worry, the cosine distance between them is so small that they are almost the same thing.

