VICTORIA UNIVERSITY OF WELLINGTON
TE HERENGA WAKA

AIML 231/DATA 302 — Week 6

Feature Construction

Dr Bach Hoai Nguyen

School of Engineering and Computer Science

Victoria University of Wellington

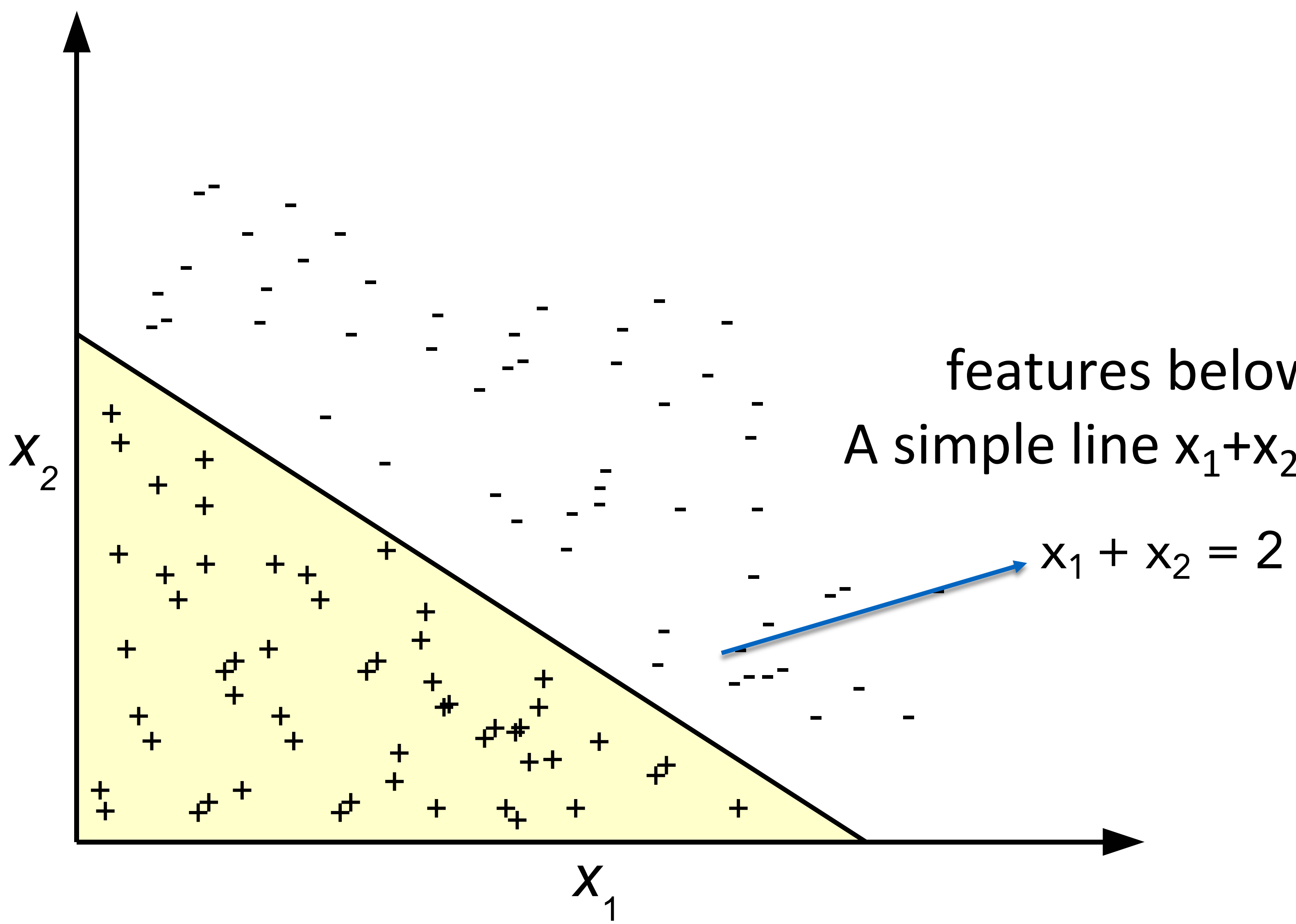
Bach.Nguyen@vuw.ac.nz

Overview

- What is Feature Construction
- Why Feature Construction
- Feature Construction Process
- Feature Construction Approaches

What is a good feature? (1)

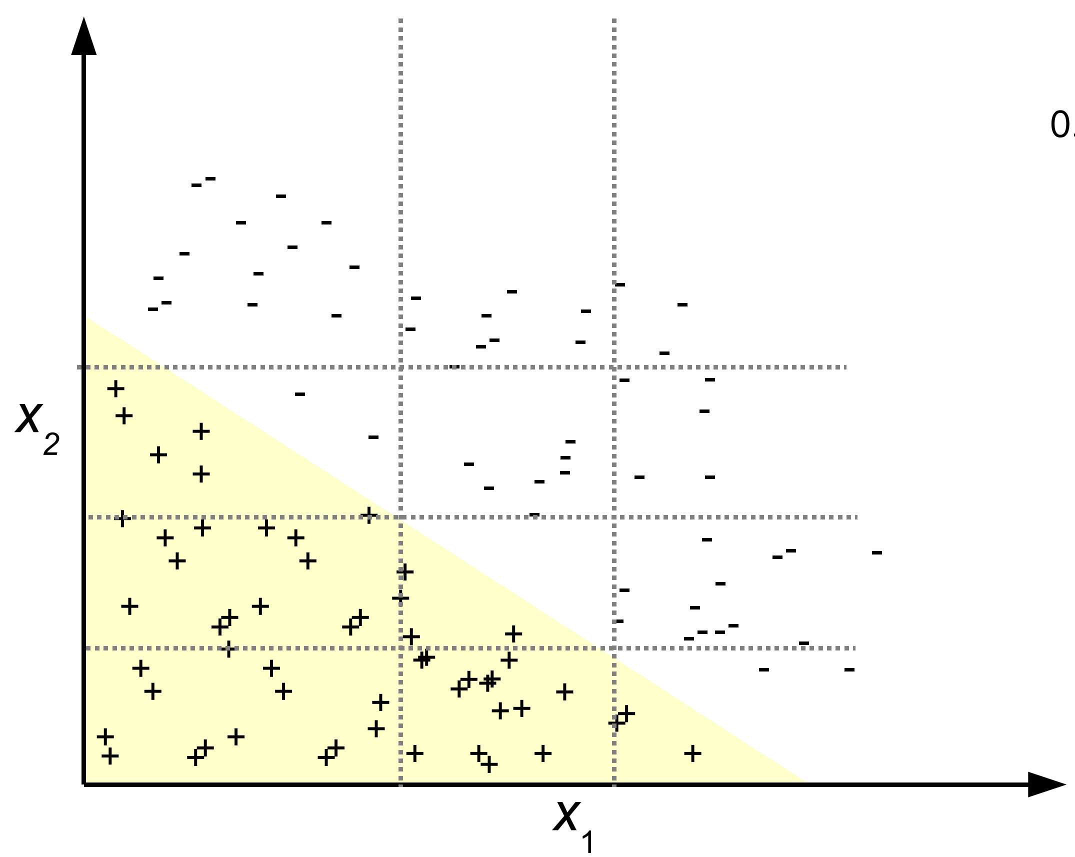
- The measure of goodness is subjective: depends on the problem.
- Example: a binary classification problem (negative class and positive class)

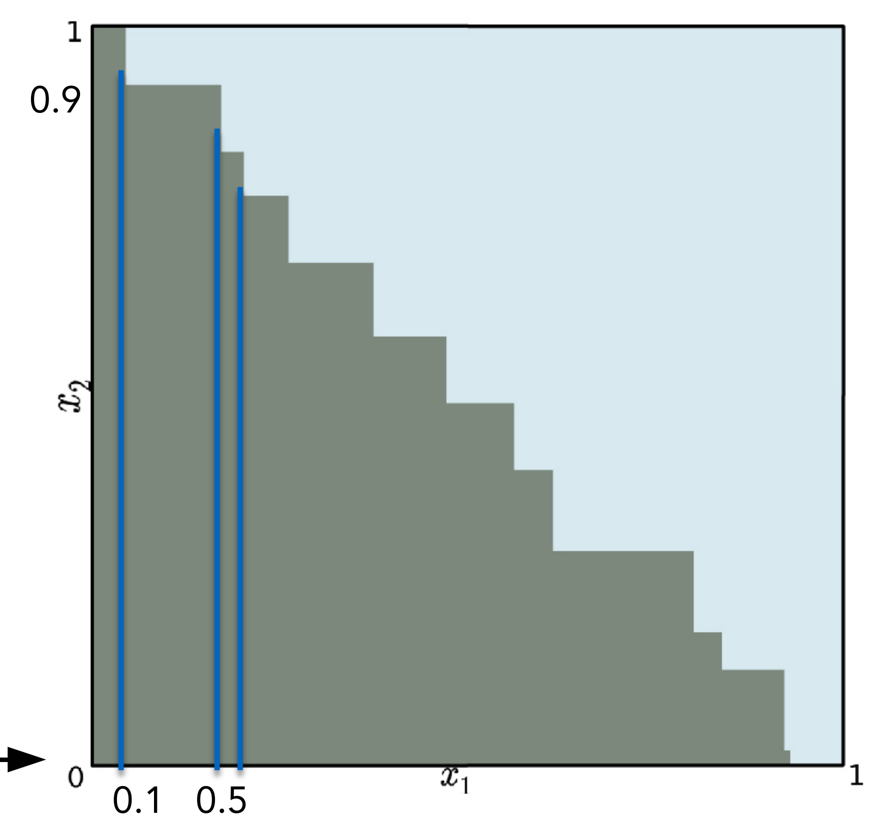


features below are good for *linear SVM*
A simple line x_1+x_2 can separate the two classes

What is a good feature? (2)

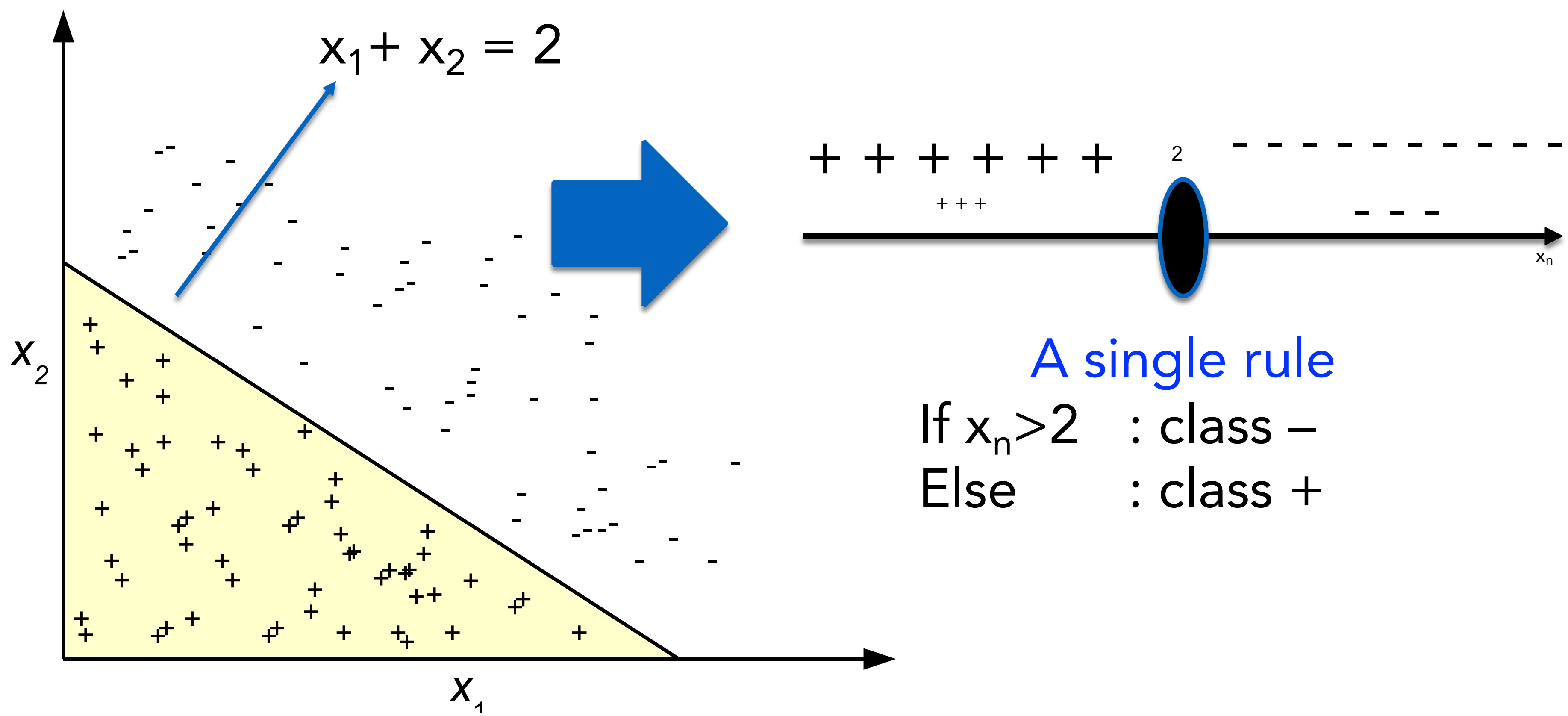
- What about Decision Tree (DT)?
- Needs to have many "Decision" rules:
  - Rule example: if $0 < x_1 < 0.1$ and $0 < x_2 < 1$ then positive class
  - Each rule is one rectangle region
- Build a large tree -> overfitting problem
- The two features are not good for a DT classifier that is not able to transform its input space





What is a good feature?

New feature: $x_n = x_1 + x_2$: constructed feature



A single rule
If $x_n > 2$: class –
Else : class +

- Linear SVM can transform the original feature space
- Decision Tree cannot transform the original feature space even if the original features have good information

Why Feature Construction?

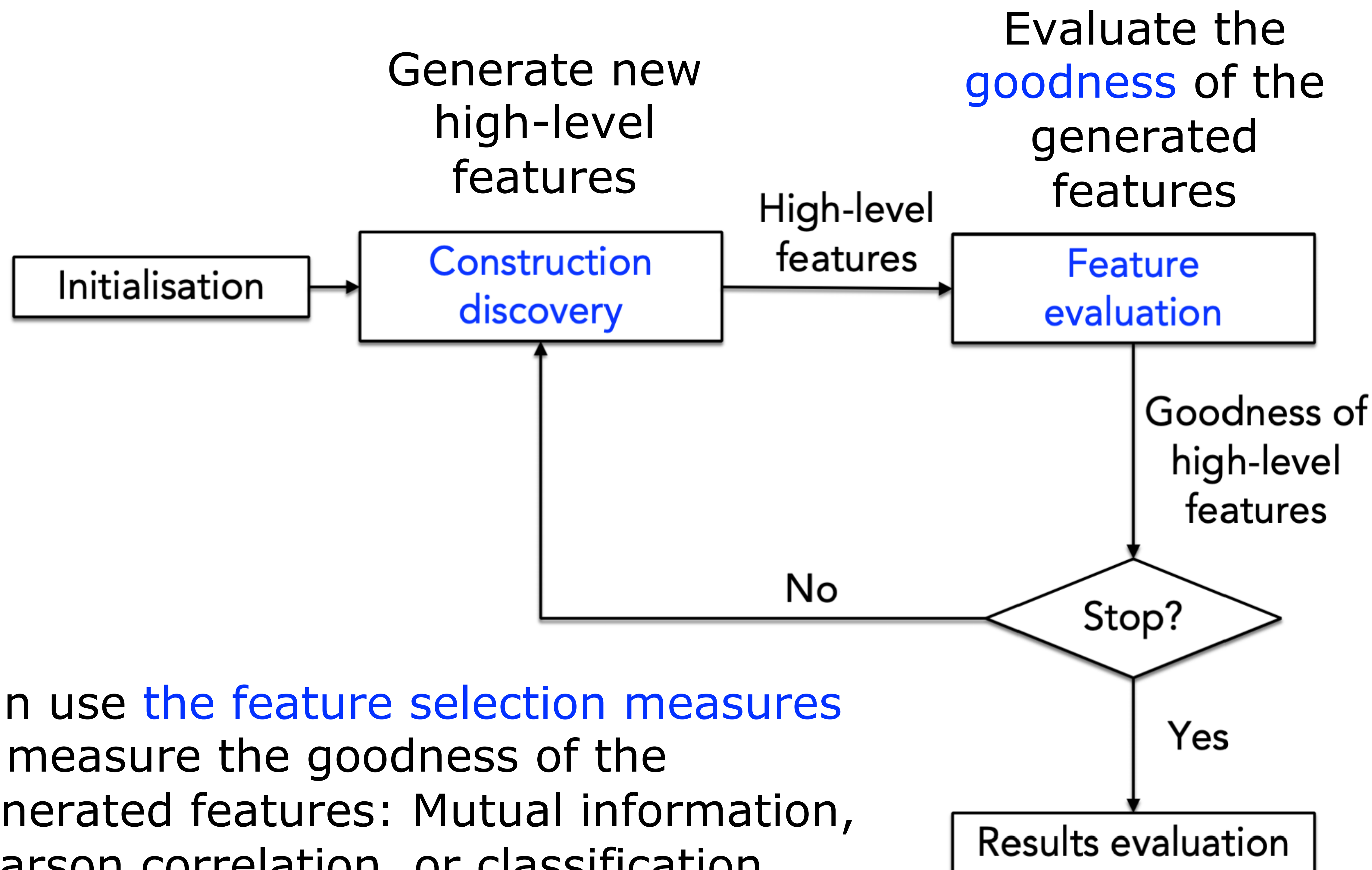
- The quality of input features can drastically affect the learning performance
- Even if the original features are high-quality, transformations may be needed to use them with certain types of classifiers
- A large number of classification algorithms are unable to transform their input space
- Feature construction does not add to the cost of acquiring original features – it only carries computational cost
- Often, feature construction can lead to dimensionality reduction or implicit feature selection

Feature Construction

- A kind of feature transformation to produce high-level constructed features that discover the relationships between features and augment the feature space
- Given $(X_1, X_2, . . . , X_m)$ - the vector corresponding to the set of original features, a constructed feature is a scalar function φ that transforms the set to a one-dimensional space: $\varphi (X_1, X_2, . . . , X_m)$

Example: Give $[X_1, X_2, X_3]$,
linear construction: $Xc=X_1+X_2$, $Xc=4X_1+3X_2+6X_3$, …
nonlinear construction: $Xc=X_1*X_2$, $Xc=X_2*X_3^2$, …
…

Feature Construction Process



- Can use the feature selection measures to measure the goodness of the generated features: Mutual information, Pearson correlation, or classification performace

Construct New Features - Operators

The choice of operators/functions is based on domain knowledge and the type of features

- Boolean features: Conjunctions, Disjunctions, Negation
- Nominal features: Cartesian product, M of N etc.
- Numerical features: Min, Max, Addition, Subtraction, Multiplication, Division, Average, Equivalence, Inequality etc.

A major challenge in feature construction: choosing the right set of operators and applying them appropriately

- Search space in feature construction is the space of possible functions of input features
  - How big is it?

Evaluate and Select New Features

- Not all constructed features are good
- Apply feature selection techniques to remove redundant and irrelevant features
- Require an effective measure to evaluate the new features and provide an indicator
- Not computationally expensive
- Measures of consistency, distance, learning performance

Feature Construction Methods

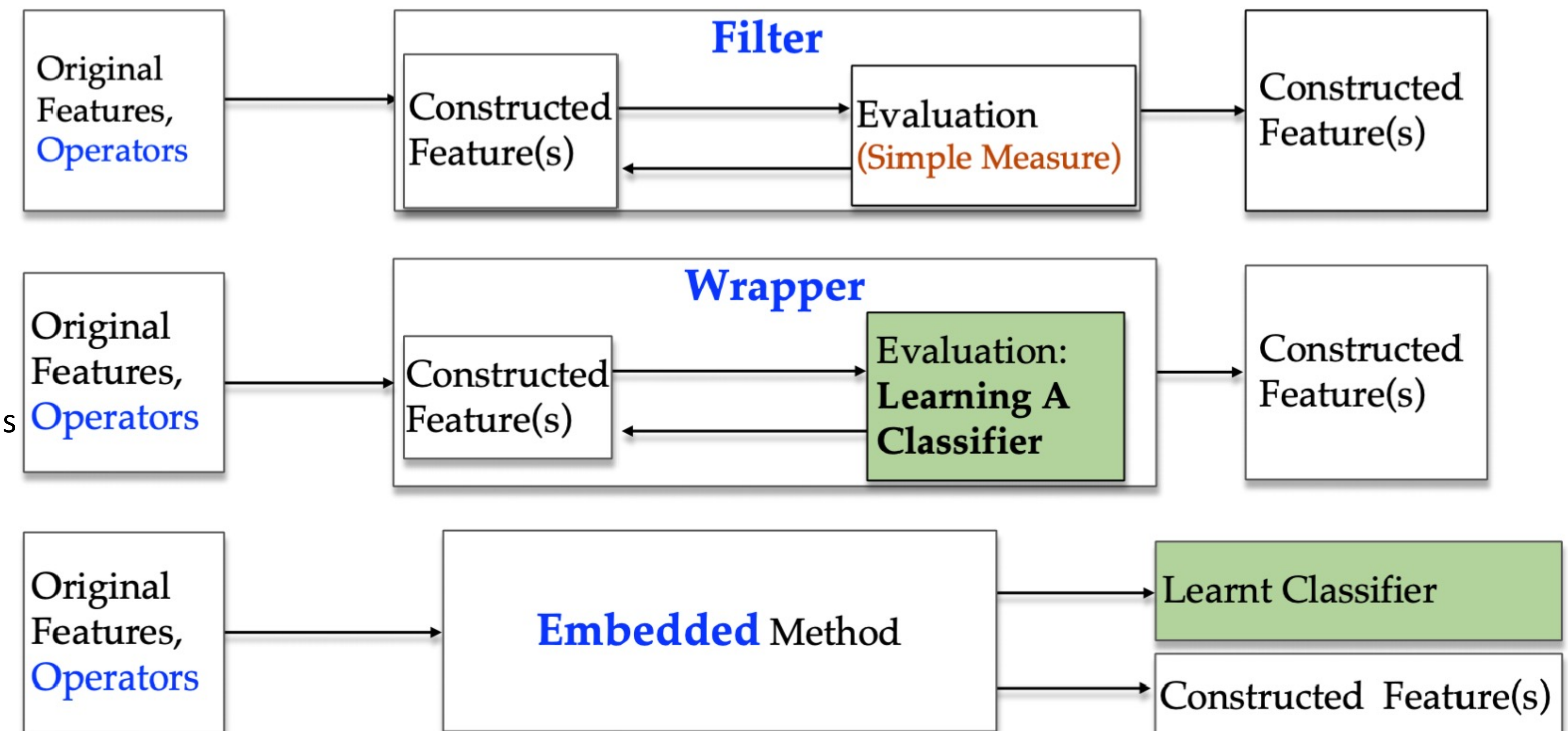
Based on Evaluation ——— learning algorithm

- Three categories: Filter, Wrapper, Embedded
- Hybrid (Combined)

Filter
Uses existing measures, no learning algorithm

Wrapper
Uses learning performance, train ML models **many times**

Embedded
Train ML model **once**
New features based on the learned model



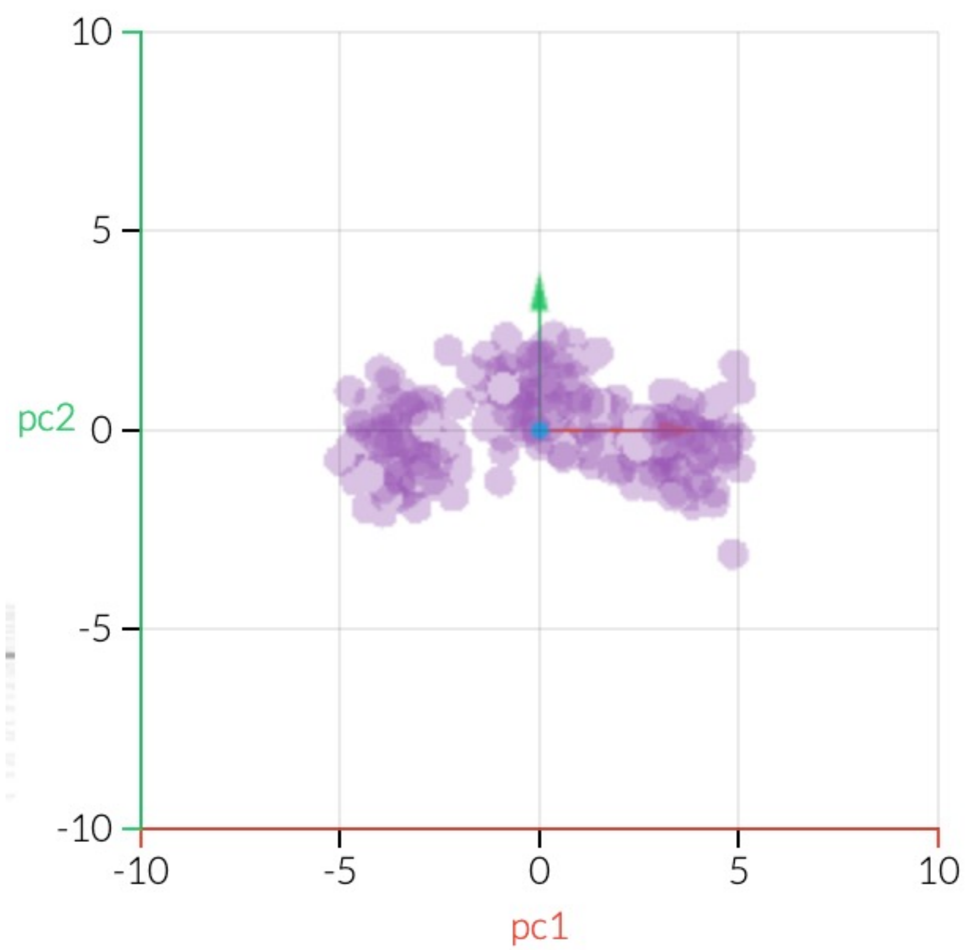
Feature Construction Methods

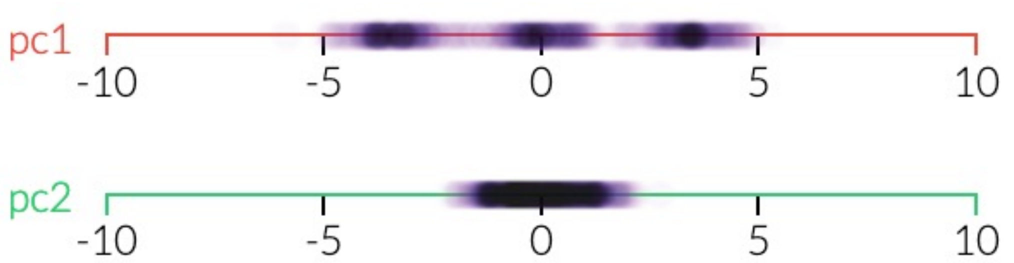
- Comparing the three categories:

	Classification Accuracy	Computational Cost	Generality (to different classifiers)
Filter	Low	Low	High
Embedded	Medium	Medium	Medium
Wrapper	High	High	Low

Principal Component Analysis (PCA)

- Invented by Karl Pearson (1901)
- *PCA* is a mathematical procedure that ***linearly*** transforms *(possibly) correlated* features into a (*smaller*) number of *uncorrelated* features called *principal components*.
- Goal is to achieve *high data variance*





- Data variance is related to the distances between data points
- Higher variance: easier to extract knowledge (for example building a classifier to separate different classes)
- pc1 has higher variance than pc2

Principal Components

From P original features: $x_1,x_2,\ldots,x_p$:

Produce K new features: $y_1,y_2,\ldots,y_k$:

$$\left.\begin{aligned}
y_1 &= a_{11}x_1 + a_{12}x_2 + \ldots + a_{1p}x_p \\
y_2 &= a_{21}x_1 + a_{22}x_2 + \ldots + a_{2p}x_p \\
&\ldots \\
y_k &= a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kp}x_p
\end{aligned}\right\}$$

y_k's are Principal Components

y_k's are uncorrelated (orthogonal)

- At most P principal components can be built
- Rank principal components based on their explained variance ratio then select K top-ranked ones
- How to set K? general rule is to preserve at least 95% data variance

Comments on PCA

- Advantages:
  - Ensure new features (principal components) are uncorrelated
  - Interpretability: the new features are easy to understand
  - Efficiency: fast to run and scale well with large datasets

- Limitations:
  - Assume a linear mapping from the original features to the new features
  - Need to define the number of new principal components
  - Sometimes uncorrelated features are not sufficient

Independent Component Analysis (ICA)

- Formally introduced by Pierre Comon in 1994

- Like PCA, ICA creates new components that are linear combinations of the original variables
- Components are as **statistically independent**

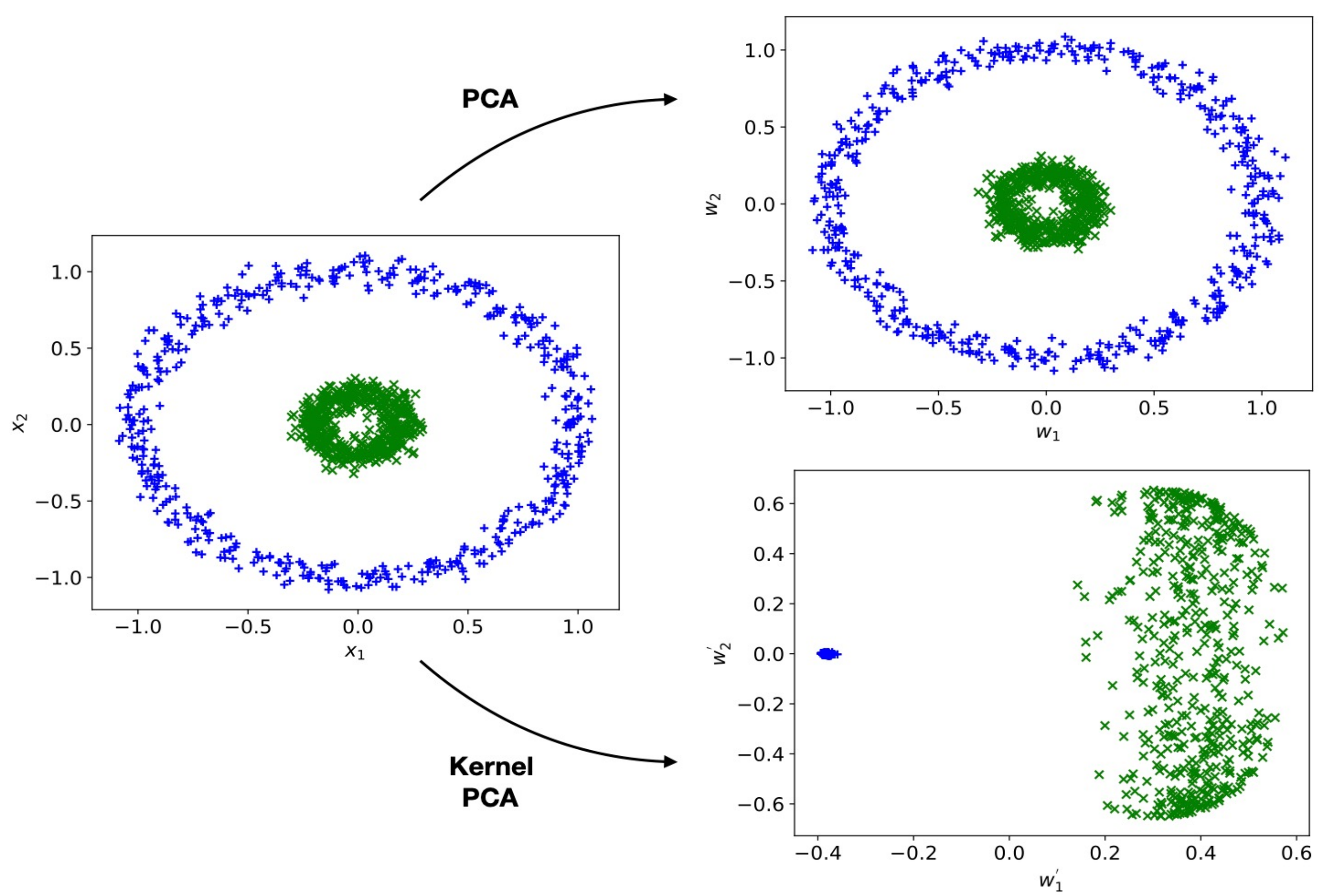
$$p(x, y) = p(x)p(y)$$

- Statistically independent is generally stronger than uncorrelated condition in PCA

- Searching for independent components by
  - Maximising the non-Gaussianity: more than just correlation
  - Minimising the statistical independence between new components

Kernel Principal Component Analysis

- Kernel PCA combines a specific mathematical view of PCA with kernel functions, nonlinear extension of PCA
- Perform PCA on the transformed data $\Phi(x)$- "blessing of dimensionality"
- Φ - Radial basis function kernel (RBF), Polynomial function, …



https://ml-lectures.org/docs/structuring_data/ml_without_neural_network-2.html

Comments on Kernel PCA

- Implicit mapping hidden behinds the kernel function
- Not as interpretable as standard PCA
- Need to define the kernel function and the parameters for the kernel function

Polynomial Features

- Polynomial features are created by raising existing features to an exponent.
- Generate a new feature matrix with the polynomial combinations of the features with a degree (less than or equal to the specified degree)

Example: Two-dimensional input feature space $[x_1, x_2]$
the ***degree-2*** polynomial features: $[1, x_1, x_2, x_1^2, x_1x_2, x_2^2]$

- Higher degree leads to larger number of features -> consider feature selection
- Change the probability distribution of features by separating the small and large values
- Appropriate degree can improve ML algorithms typically linear algorithms

GP For Feature Construction (Bonus)

- Genetic Programming is flexible in making mathematical and logical functions
- There isn't much structural (topological) information in the search space of possible functions, so using a meta-heuristic approach (such as evolutionary computation) seems reasonable