Big Data

AIML427

Feature Construction

Dr Bach Hoai Nguyen

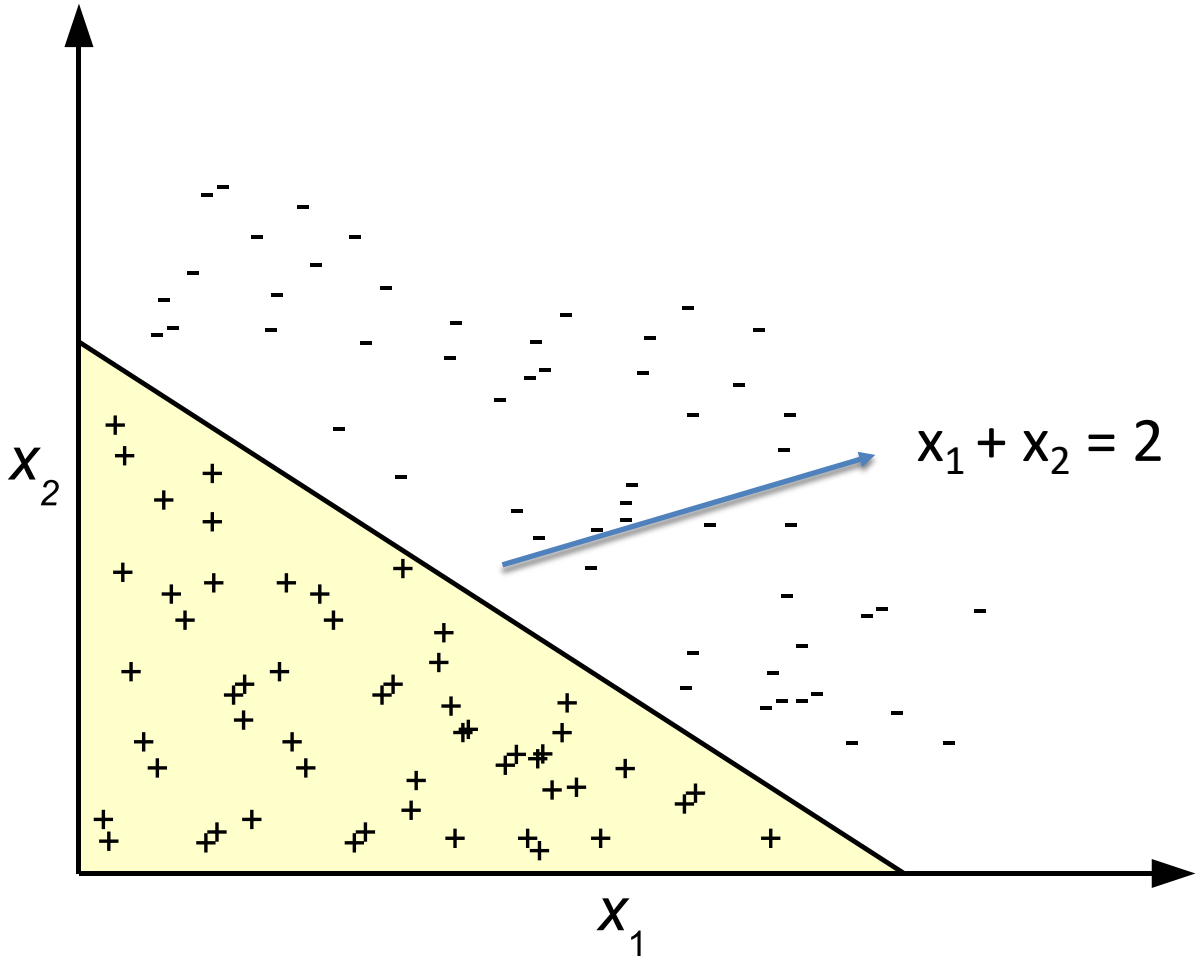
Bach.Nguyen@vuw.ac.nz

Outline

- Feature Construction
  - What is feature construction?
  - Why do feature construction?
  - Challenges in feature construction
  - Feature construction process
  - Principle component analysis (PCA)
  - Genetic Programming for Feature Construction
- Issues and Challenges in FS and FC
- Summary

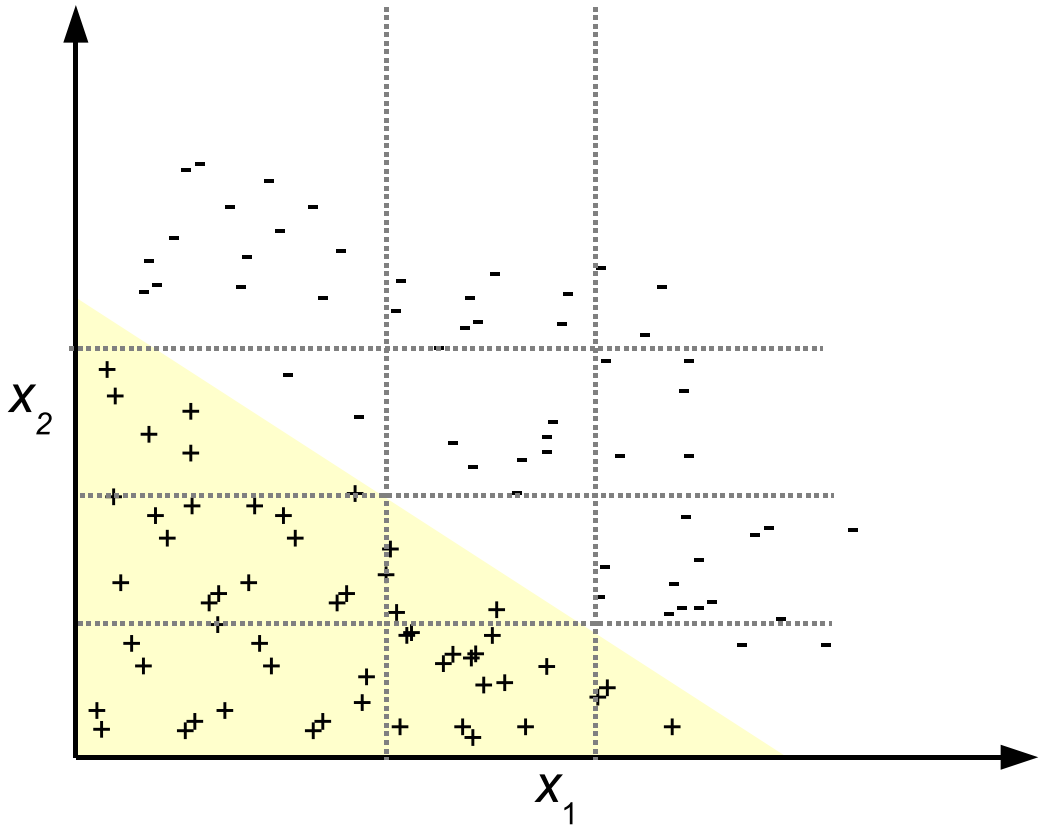
What is a Good Feature?

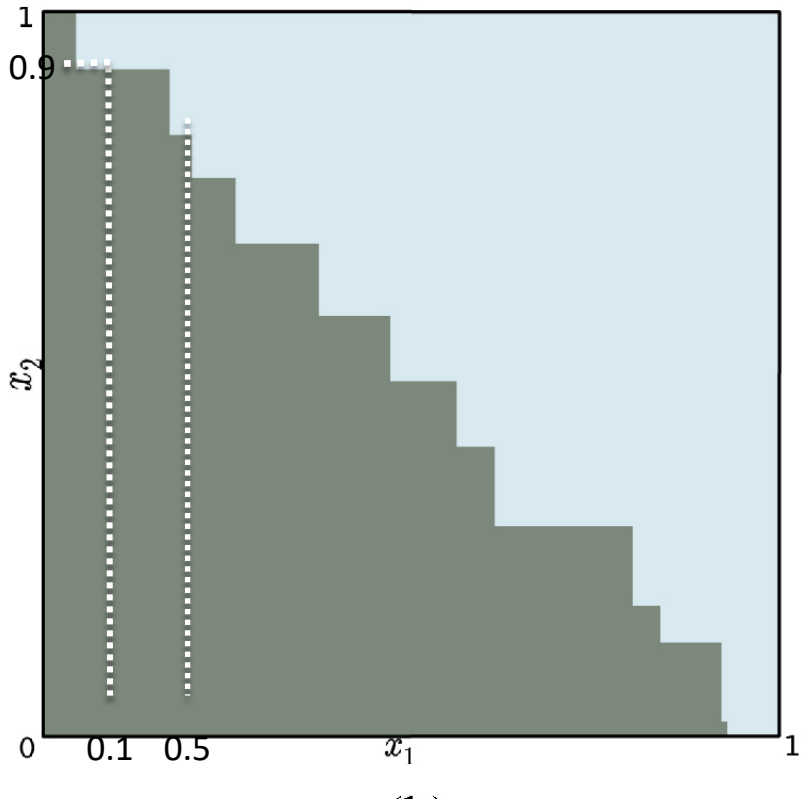
- The measure of goodness is subjective: depends on the problem. The features below are good for a *linear classifier*



What is a Good feature?

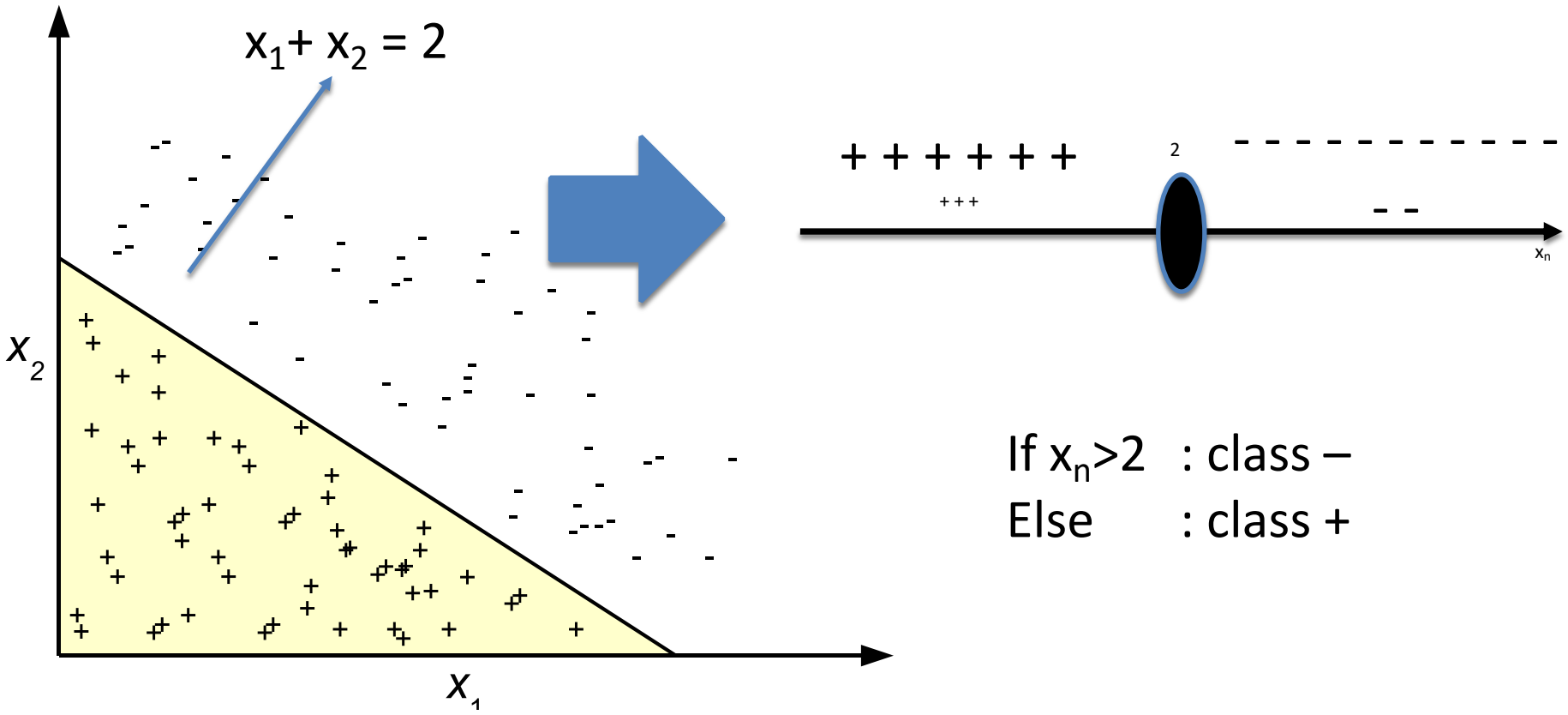
- The same set of features are not good for a DT classifier that is not able to transform its input space





What is a Good feature?

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



If $x_n > 2$: class –

Else : class +

Why Do We Do Feature Selection?

- **"Curse of dimensionality"**
  - Large number of features: 100s, 1000s, even millions
- Not all features are useful (relevant)
- Redundant or irrelevant features may reduce the performance (e.g. **classification accuracy**). Confuse many learning algorithms. How?
  - Example 1: decision tree learning
  - Example 2: Bayesian learning

- Costly: time, memory, and money

Why Do 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

Definitions:

1. A constructed feature is a scalar ***function*** ϕ that ***transforms*** the input space to a one-dimensional space

2. Given a feature vector (X1, X2, . . . , Xm), a constructed feature is a function of the form ϕ(X1, X2, . . . , Xm)

3. The term Feature Construction refers to the process of producing high-level constructed features

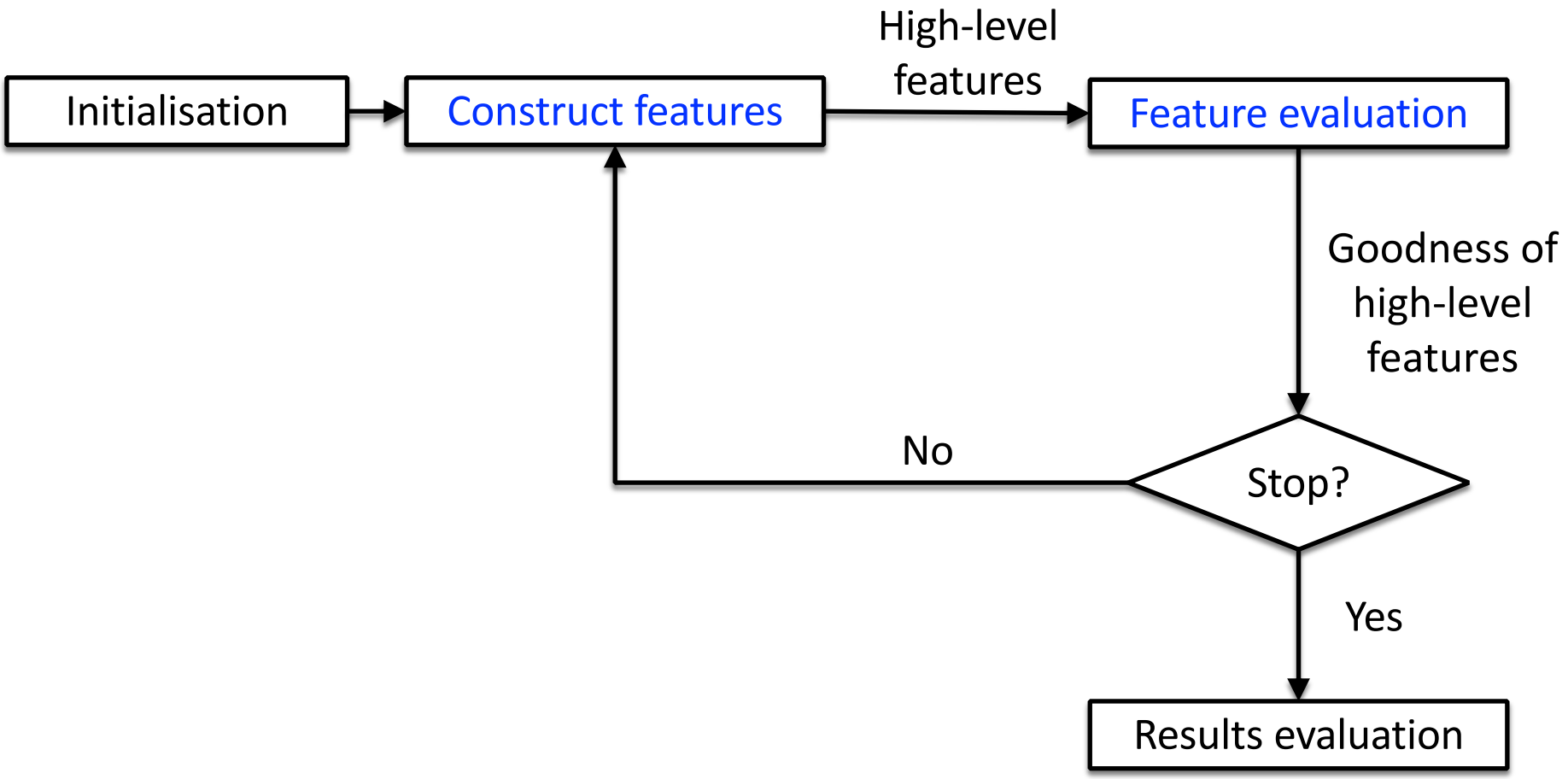
4. Feature construction =
   Feature selection + Combining selected features

Challenges in Feature Construction?

- Challenges in Feature Selection?

- Additional challenges:
  - when to perform feature construction?
  - even bigger search space (how big…?)
  - Easier to overfit?

Feature Construction Process



Components of a Feature Construction System

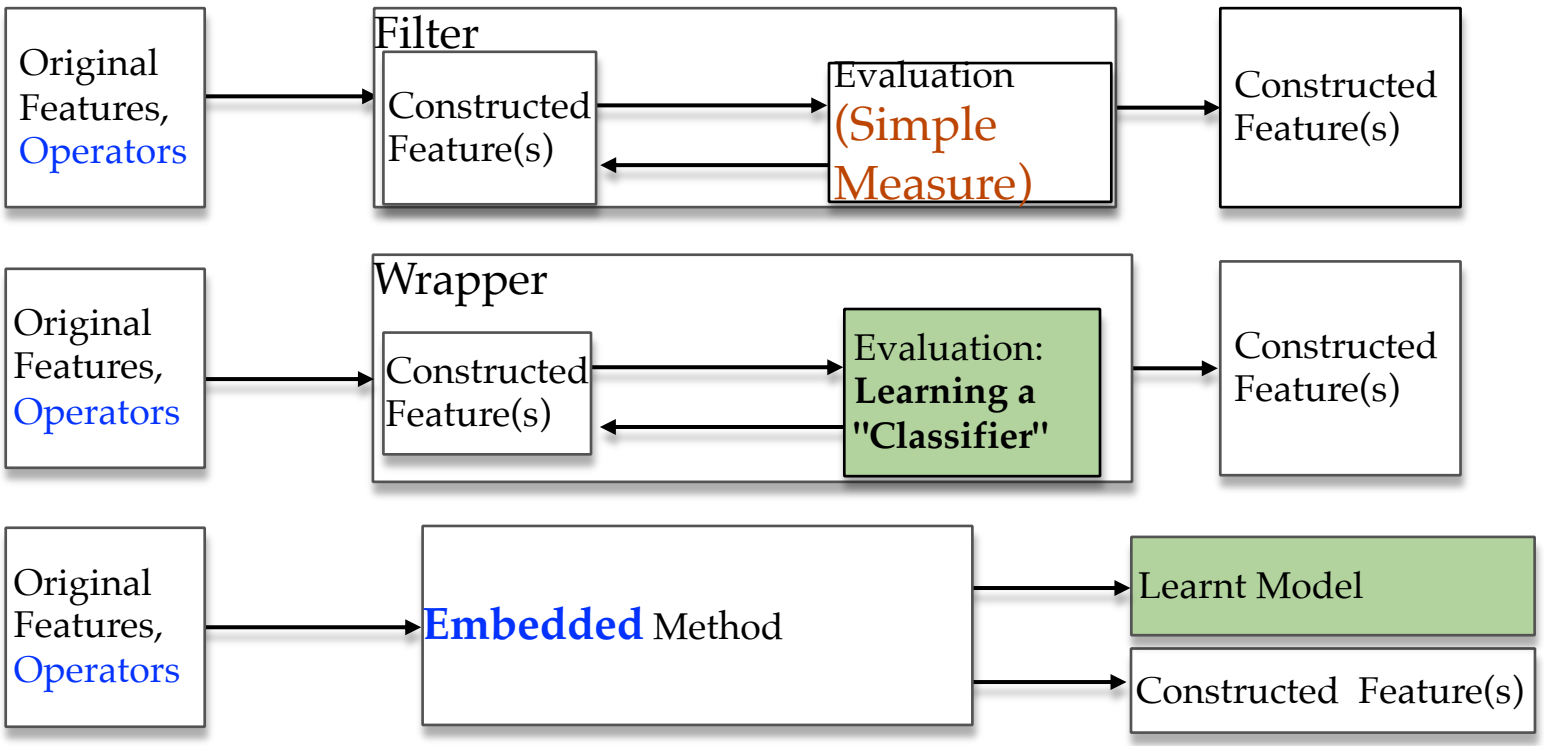
- In feature construction the search space is the space of possible functions of input features
  - E.g. $Feature_{n+1} = Feature_1 + Feature_2 \times Feature_3 - Feature_4$

A typical FC system has the following two components:

- A *search strategy* to search the space of possible functions (of original features)
- An *evaluation mechanism* to measure the goodness of a candidate function

Feature Construction Approaches

- Based on how the feature subset is evaluated
  - Three categories: Filter, Wrapper, Embedded
  - Hybrid (Combined)



Feature Construction Approaches

- Same as feature selection!

- Generally:

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

PRINCIPAL COMPONENT ANALYSIS (PCA)

Variance and Covariance

- Variance measures how spread out a feature is:

$$var(X) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1}$$

- Higher variance → more information

- Covariance measures the relationship between two features:

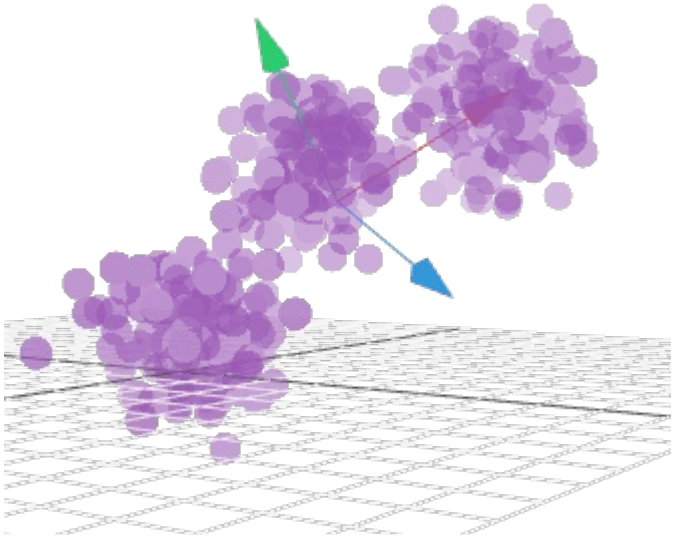
$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{n-1}$$

- In data analysis, we want features to be independent

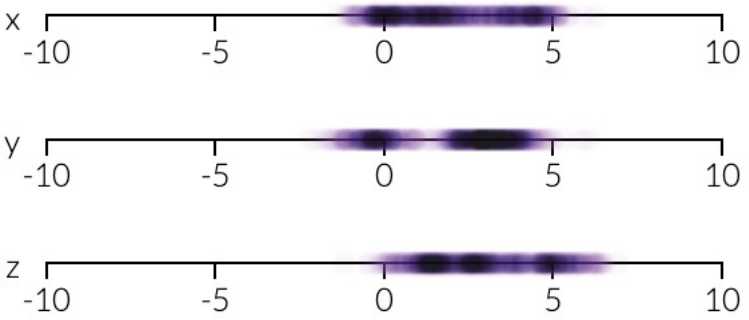
- **We want high variance and low covariance**

Principal Component Analysis

- Original data

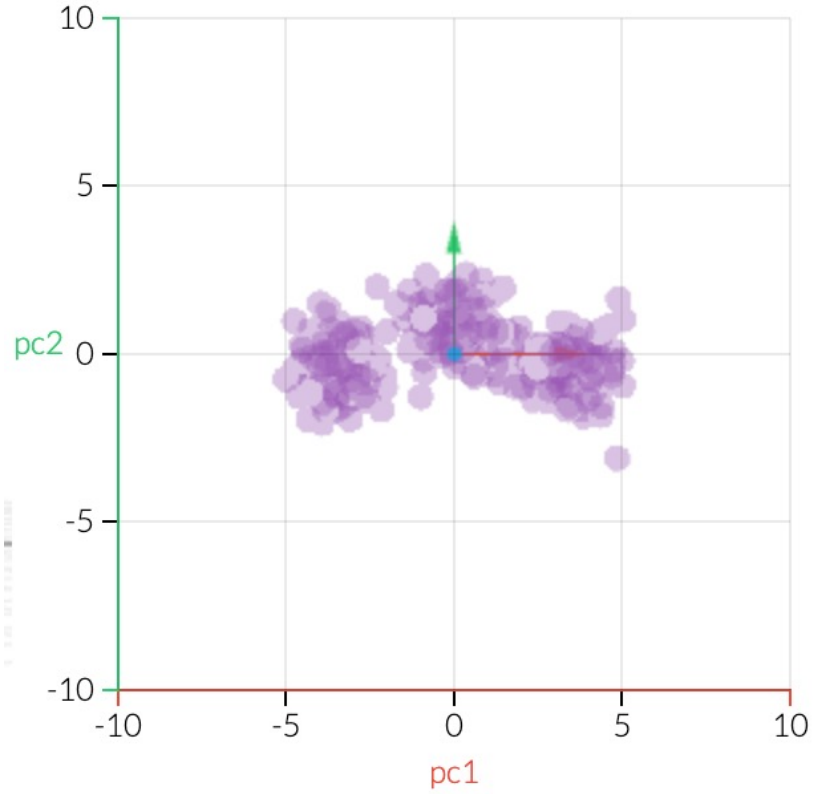


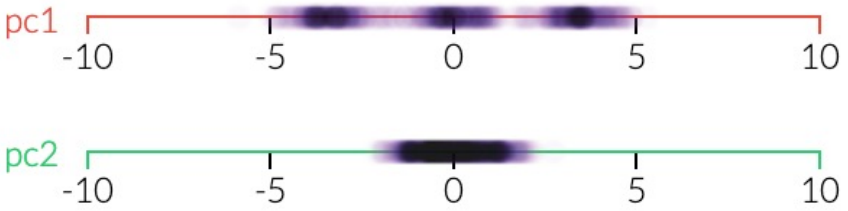
(x,y,z)



Principal Component Analysis

- Transformed data using PCA





Principal Component Analysis

- Invented by Karl Pearson (1901): the most widely-used and well-known of the "standard" *multivariate* methods

- *PCA* is a mathematical procedure that transforms *(possibly) correlated* variables into a (*smaller*) number of *uncorrelated* variables called *principal components*.

- *Summarisation* of data with many (*p*) variables by a *smaller* set of (*k*) *derived (synthetic, composite)* variables.

Principal Component Analysis

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

Produce k new variables: $y_1, y_2, \ldots, y_k$:

$$y_1 = a_{11}x_1 + a_{12}x_2 + \ldots + a_{1p}x_p$$
$$y_2 = a_{21}x_1 + a_{22}x_2 + \ldots + a_{2p}x_p$$
$$\ldots$$
$$y_k = a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kp}x_p$$

y_k's are Principal Components

such that:

y_ks are uncorrelated (orthogonal)

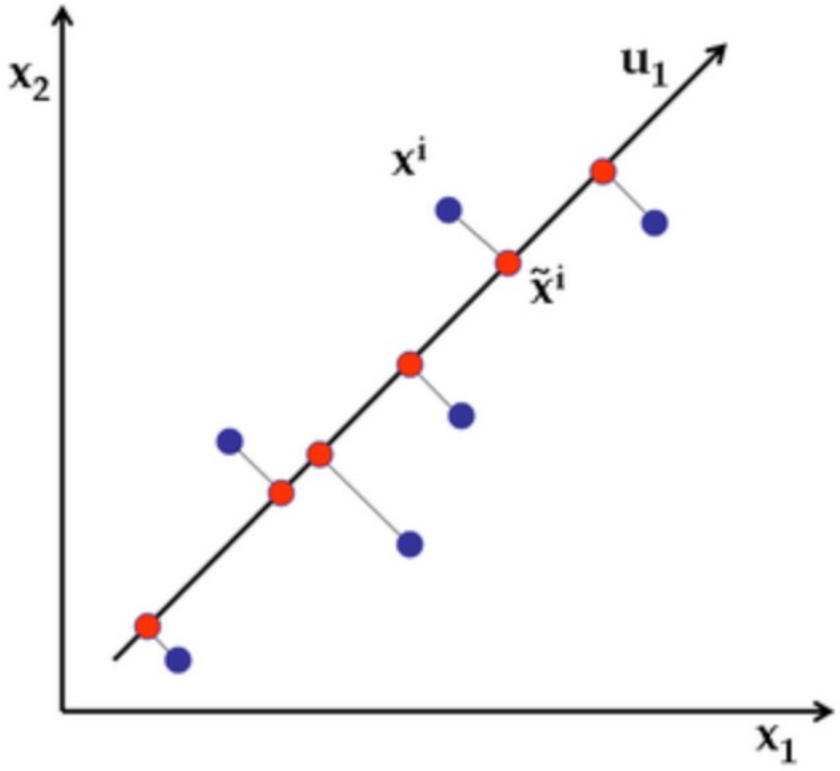
y_1 explains as much as possible of original variance in dataset

y_2 explains as much as possible of remaining variance, etc.

Principlal Component Analysis

- PCA finds a *linear projection* of *high* dimensional data into a *lower* dimensional subspace such that:
  - The *variance* retained is *maximised*.
  - The *least square reconstruction error* is *minimised*.

- Balancing act between
  - *clarity* of representation, ease of understanding
  - *oversimplification*: loss of important or relevant information.



PCA Steps

PCA steps (to reduce dimensionality from *p* to *k*):

1. Centre the data (subtract the mean)
2. Calculate the *p*p* covariance matrix
3. Calculate the *p eigenvectors* of the *covariance* matrix (*orthogonal*)
4. Select the *k eigenvectors* that correspond to the *highest k eigenvalues* to be the new space dimensions.
   - The *variance* in each new dimension is given by the *eigenvalues*
   - *How* to select *k*? Look for prominent gap in the sorted eigenvalues

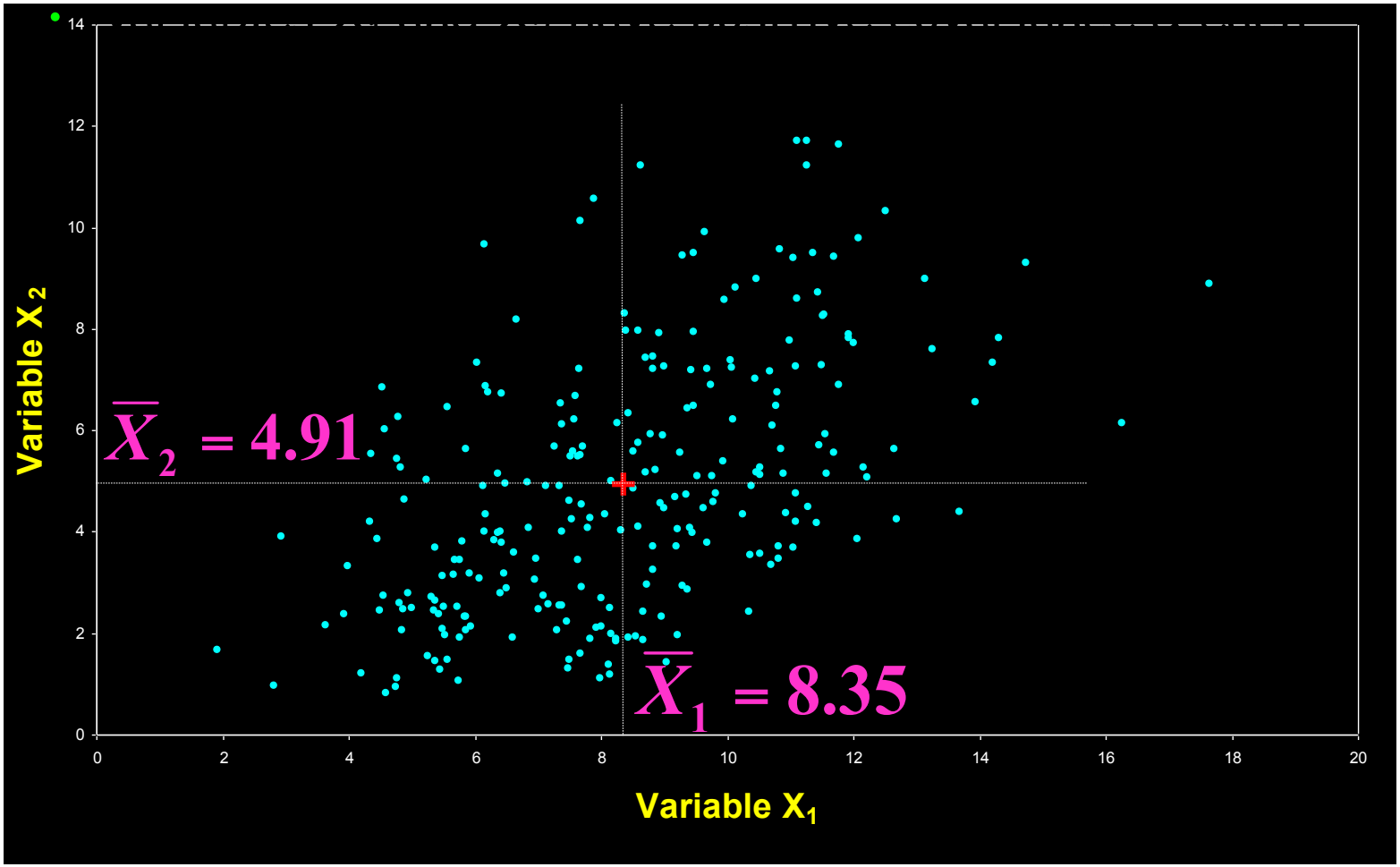
PCA: Centre the Data

- Instances are represented as a cloud of ***n*** points in a multidimensional space with an axis for each of the ***p*** variables
- the ***centroid*** of the points is defined by the *mean* of each variable, $\bar{X}_i$
- the ***variance*** of each variable V_i is the *average squared deviation* of its ***n*** values around the *mean* of that variable

$$V_i = \frac{1}{n-1}\sum_{m=1}^{n}(X_{im} - \bar{X}_i)^2$$

2D Example of PCA

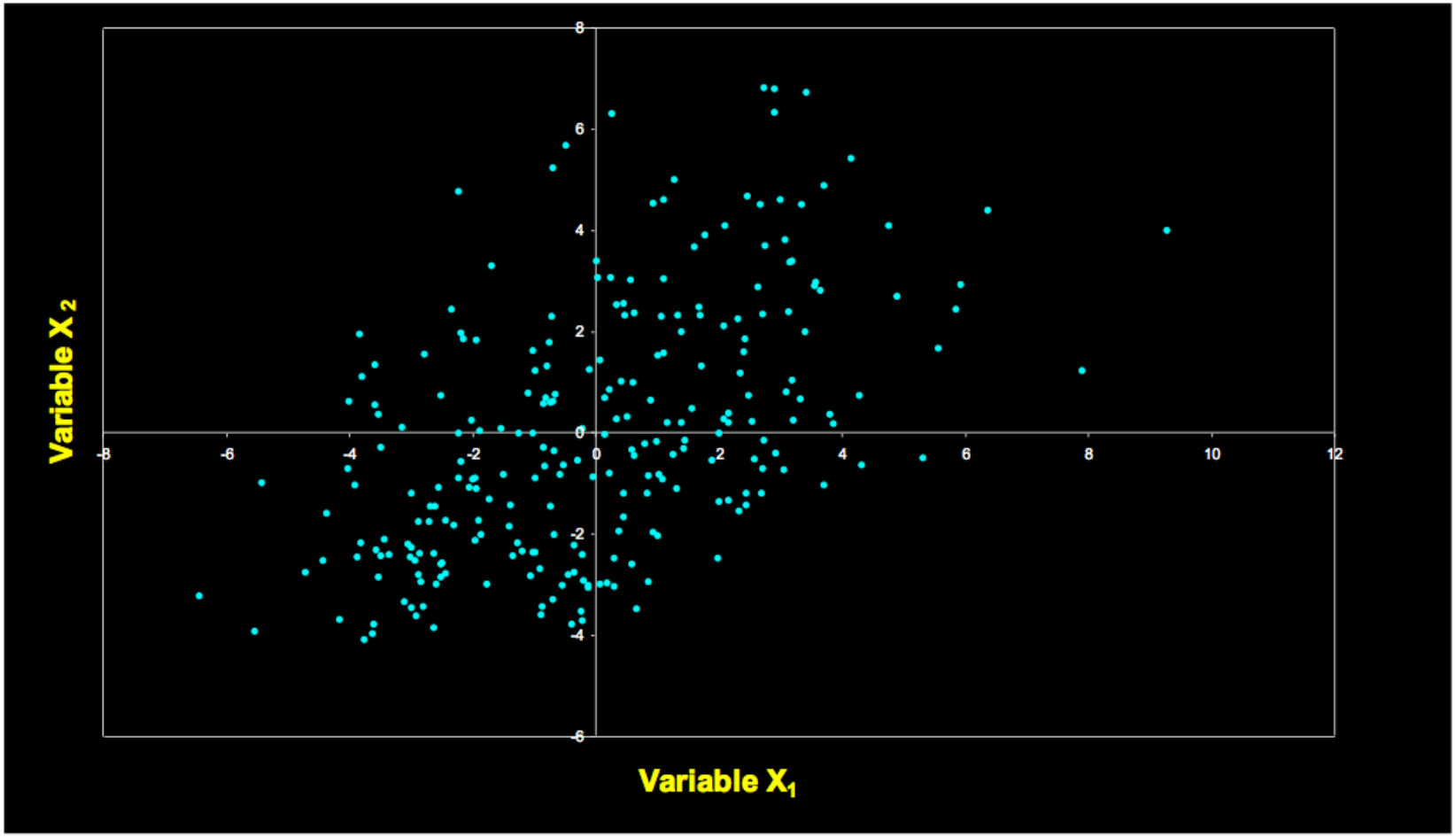
- variables X_1 and X_2 have positive covariance and each has a similar variance



$V_1 = 6.67, \quad V_2 = 6.24, \quad C_{1,2} = 3.42$

Configuration is Centred

- Each variable is adjusted to ***a mean of zero*** *(by subtracting the mean from each value)*.



PCA: Covariance matrix

- Covariance between two variables tells us how strongly they are ***linearly*** correlated.

$$C_{ij} = \frac{1}{n-1} \sum_{m=1}^{n} (X_{im} - \bar{X}_i)(X_{jm} - \bar{X}_j)$$

- C_{ij}: Covariance of variables *i* and *j*
- $\sum_{m=1}^{n}$: Sum over all *n* objects
- X_{im}: Value of variable *i* in object *m*
- $\bar{X}_i$: Mean of variable *i*
- X_{jm}: Value of variable *j* in object *m*
- $\bar{X}_j$: Mean of variable *j*

- ***p*p*** covariance matrix
- Diagonals (C_{ii}) are the variances
- Off-diagonals are the covariance
- Symmetrical along the diagonal

p = 3

$$C = \begin{pmatrix} cov(x_1, x_1) & cov(x_1, x_2) & cov(x_1, x_3) \\ cov(x_2, x_1) & cov(x_2, x_2) & cov(x_2, x_3) \\ cov(x_3, x_1) & cov(x_3, x_2) & cov(x_3, x_3) \end{pmatrix}$$

PCA: Eigenvectors & Eigenvalues

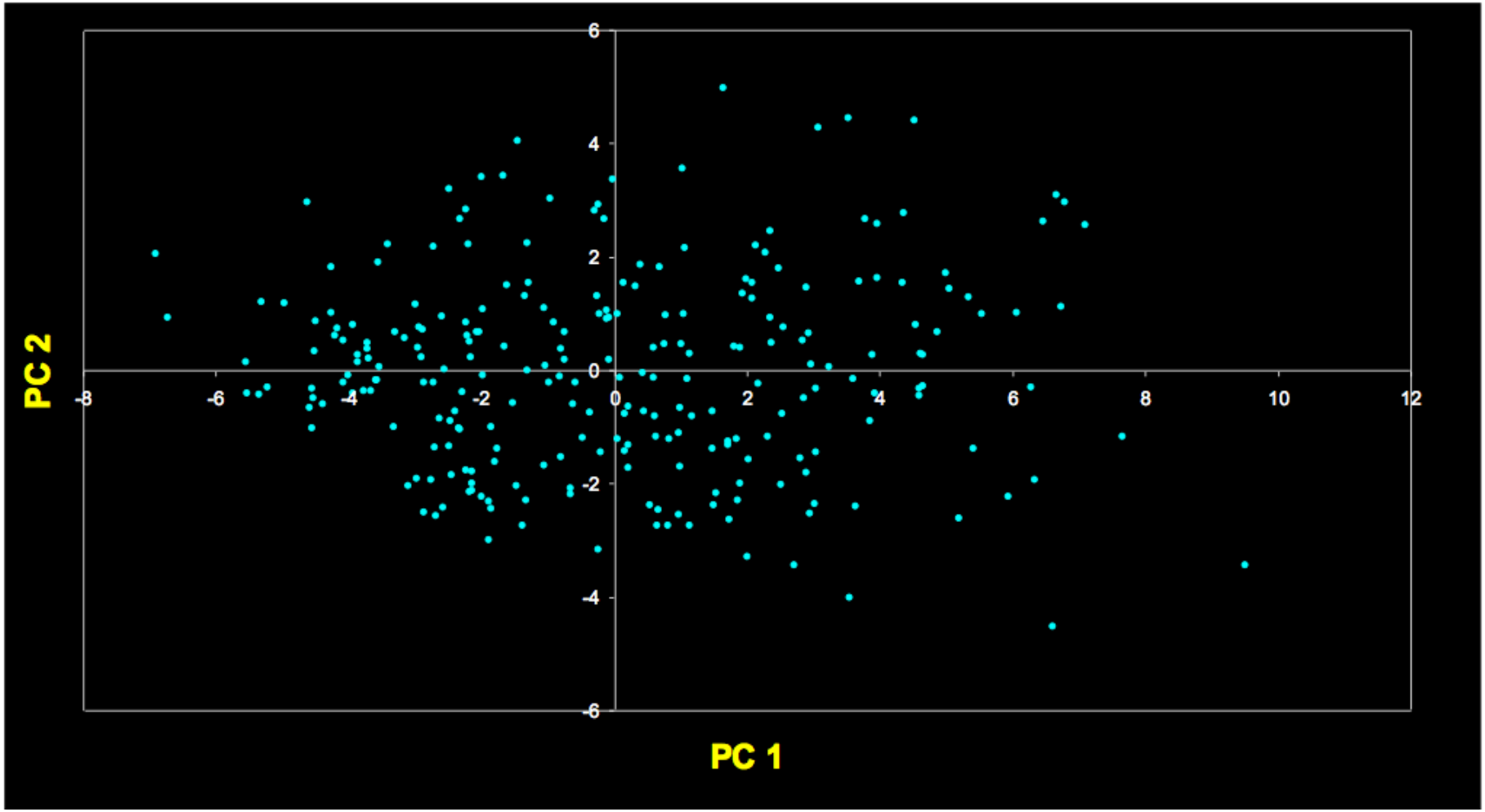
- Given a square matrix ***A*** *(p x p)*, ***x*** is its eigenvector with respect to the eigenvalue λ if: $\boldsymbol{A}^*\boldsymbol{x} = \lambda^*\mathbf{x}$
  - *p* eigenvectors and *p* corresponding eigenvalues
  - all eigenvectors are **perpendicular**

- Calculate the ***p eigenvectors*** of the ***covariance*** matrix (orthogonal), and ***p eigenvalues***
  - Each eigenvector represents a principal component
  - The corresponding eigenvalue is the variance of the component

- Select the ***k eigenvectors*** that correspond to the ***highest k eigenvalues*** to be the new space dimensions

Principal Components are Computed

- PC 1 has the highest possible variance (9.88)
- PC 2 has a variance of 3.03
- PC 1 and PC 2 have zero covariance



Generalisation to p-dimensions

- In practice nobody uses PCA with only 2 input variables

- The algebra for finding principal axes readily generalises to p variables:
  - PC 1 is the direction of maximum variance in the p-dimensional cloud of points
  - PC 2 is in the direction of the next highest variance, subject to the constraint that it has zero covariance with PC 1.
  - PC 3 is in the direction of the next highest variance, subject to the constraint that it has zero covariance with both PC 1 and PC 2
  - and so on... up to PC p

Assumptions of PCA

- Assumes relationship between variables are linear
- How many principal components (high-level features)?

- Too much to take in? A good step-by-step tutorial here: http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

Going beyond linearity

- Linear transformations are *inherently* limited
- To retain more information (variance) in fewer dimensions, we need to use non-linear transformations.

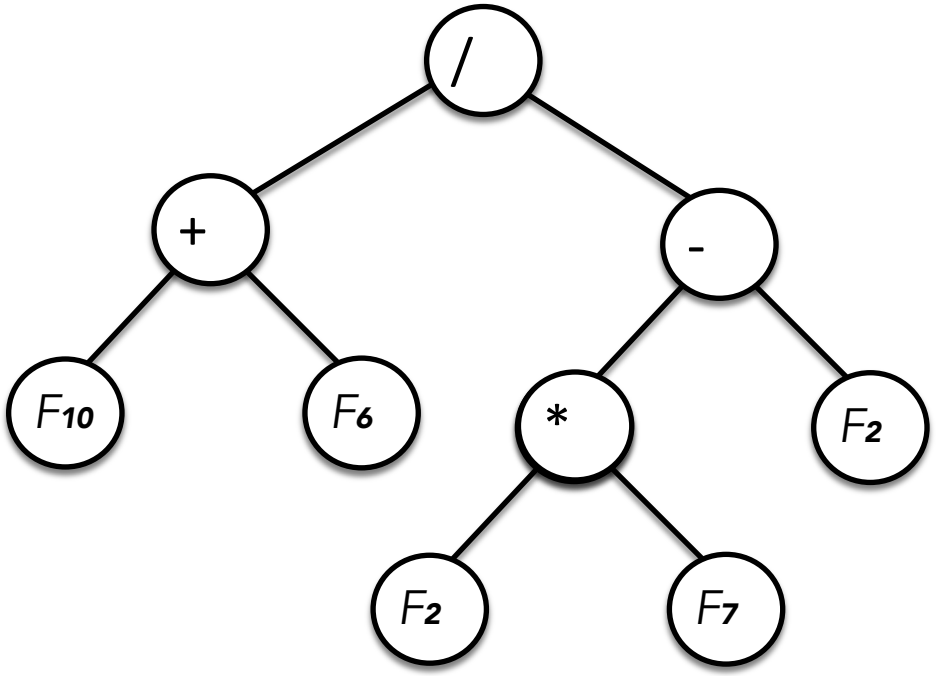
- How?

- PCA is optimal for linear transformations!
  So what is optimal for nonlinear ones?
  - Open (ill-defined?) question.
  - Today: briefly, GP for (*supervised*) feature construction
  - Next week: Manifold learning (Nonlinear dimensionality reduction) – including some of my own research ☺

Why Use GP for Feature Construction?

- Genetic programming (GP) is flexible in making mathematical and logical functions

- Not clear (or possible?) how to mathematically derive the function structure/model from the data – so a meta-heuristic approach (such as EC/GP) seems reasonable



$$(F_{10} + F_6)/(F_2 * F_7 - F_2)$$

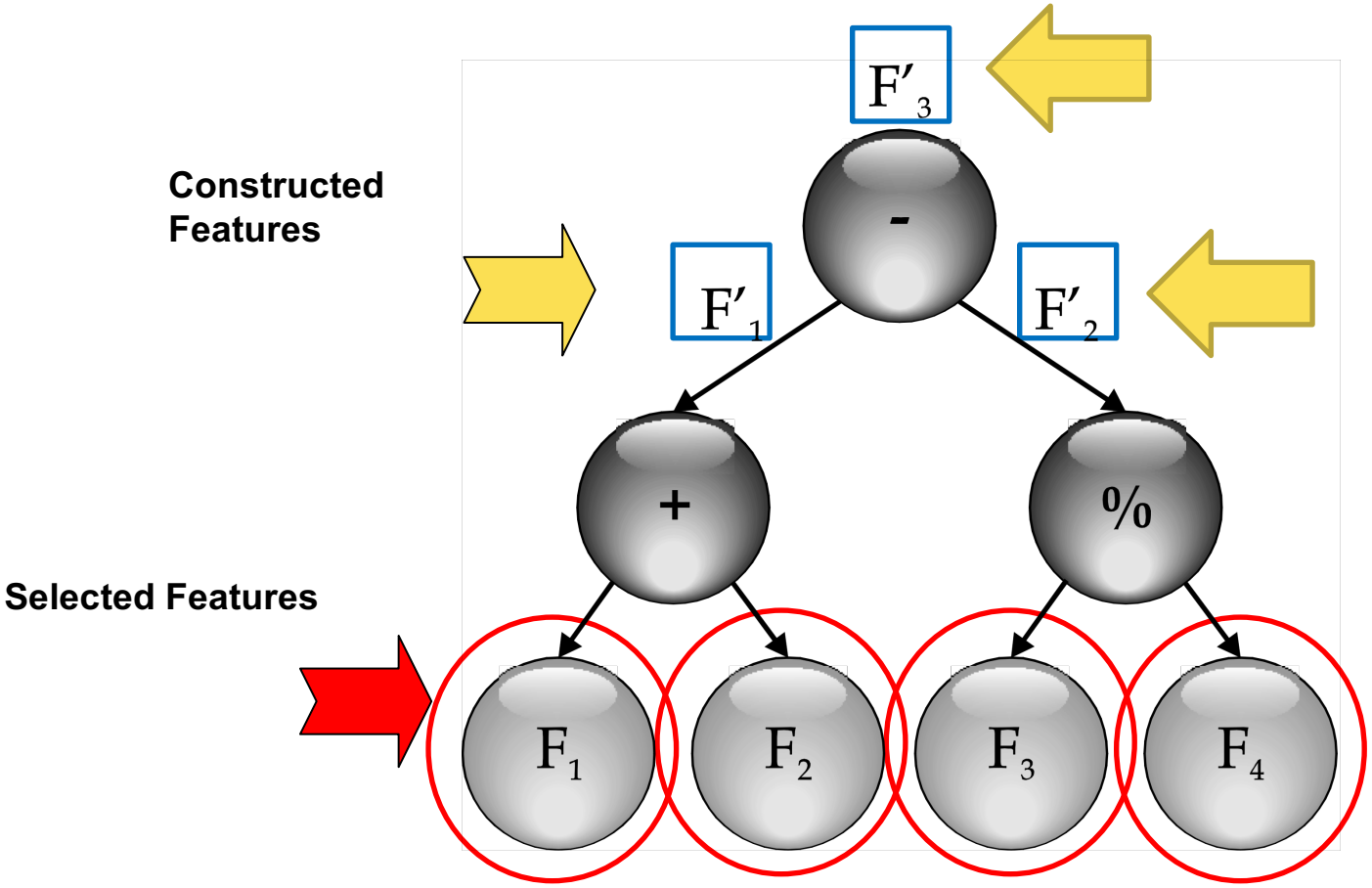
Single vs. Multiple Feature Construction

- One constructed feature – easy, evolve a single tree.
  - Often “augment” the original dataset.

- Possible ways to make multiple features are:
  - random restart and picking multiple individuals. Often leads to very high correlation between constructed features. ☹
  - Change our representation to suit!

Multiple feature construction: single-tree

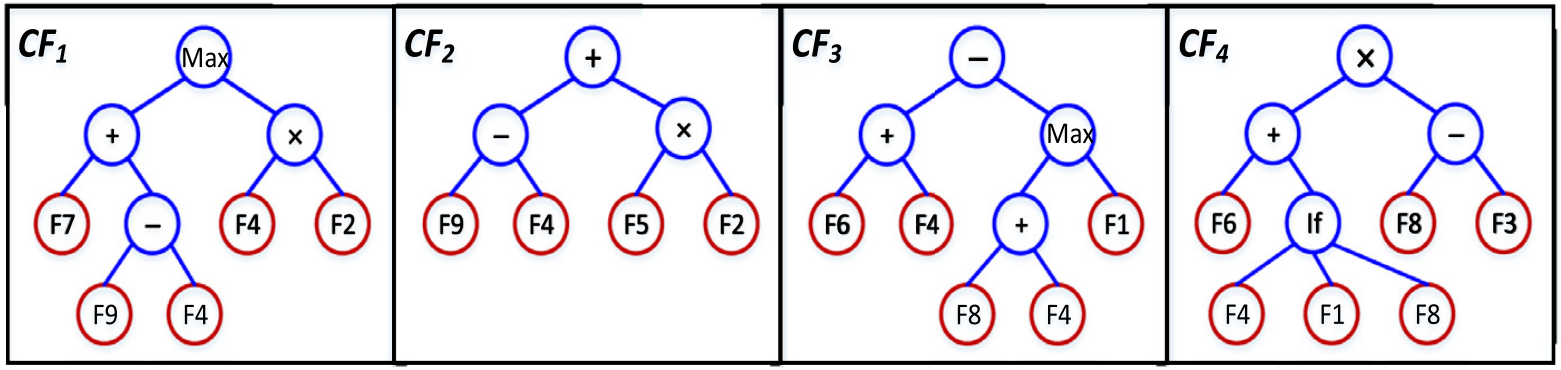
- Construct multiple features from a single tree



Soha Ahmed, Mengjie Zhang, Lifeng Peng and Bing Xue."Multiple Feature Construction for Effective Biomarker Identification and Classification using Genetic Programming". Proceedings of 2014 Genetic and Evolutionary Computation Conference (GECCO 2014). ACM Press. Vancouver, BC, Canada. 12-16 July 2014.pp.249--256

Multiple feature construction: multi-tree (1)

- Each individual is a set of trees (*m trees*)
- Class-independent: $m = r \times c$
  - e.g. 2 classes ($c=2$), $r = 2$ -> 4 trees in each individual

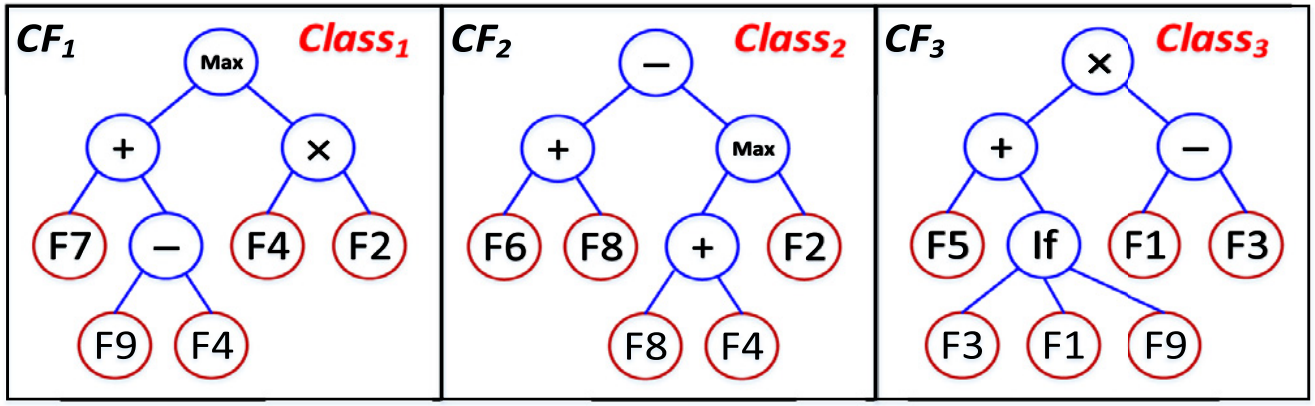


- $CF_1 = Max(F7 + F9 – F4, F4 \times F2)$
- $CF_2 = (F9 – F4) + (F5 \times F2)$
- $CF_3 = (F6 + F4) – Max(F8 + F4, F1)$
- $CF_4 = (F6 + If\ (F4, F1, F8)) \times (F8 – F3)$

Binh Tran and Bing Xue and Mengjie Zhang. "Genetic programming for multiple-feature construction on high-dimensional classification", Pattern Recognition, vol. 93, pp. 404-417, 2019.

Multiple feature construction: multi-tree (2)

- Class-dependent: $m = r \times c$
  - e.g. 3 classes ($c=3$), $r = 1$ -> 3 trees in each individual



- CF_1: distinguish between $Class_1$ and other classes
- CF_2: distinguish between $Class_2$ and other classes
- CF_3: distinguish between $Class_3$ and other classes

Binh Tran and Bing Xue and Mengjie Zhang. "Genetic programming for multiple-feature construction on high-dimensional classification", Pattern Recognition, vol. 93, pp. 404-417, 2019.

GP-based feature construction

- Multiple-feature construction > single-feature construction

- Among multiple-feature construction:
  - Multi-tree representation > single-tree representation

- Among multi-tree representation:
  - Class-dependent > class-independent

Binh Tran and Bing Xue and Mengjie Zhang. "Genetic programming for multiple-feature construction on high-dimensional classification", Pattern Recognition, vol. 93, pp. 404-417, 2019.

Issues and Challenges in FS and FC

- Scalability Problem
  - thousands, tens of thousands, and even millions
- Computational Cost
- Search Mechanisms
- Measures
- Representation
- Multi-Objective Feature Selection
- Number of Instances
- Stability

Feature Selection/Construction Approaches

- Generally:

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