Big Data

AIML427
Feature Manipulation

Dr. Bach Hoai Nguyen

Bach.Nguyen@vuw.ac.nz

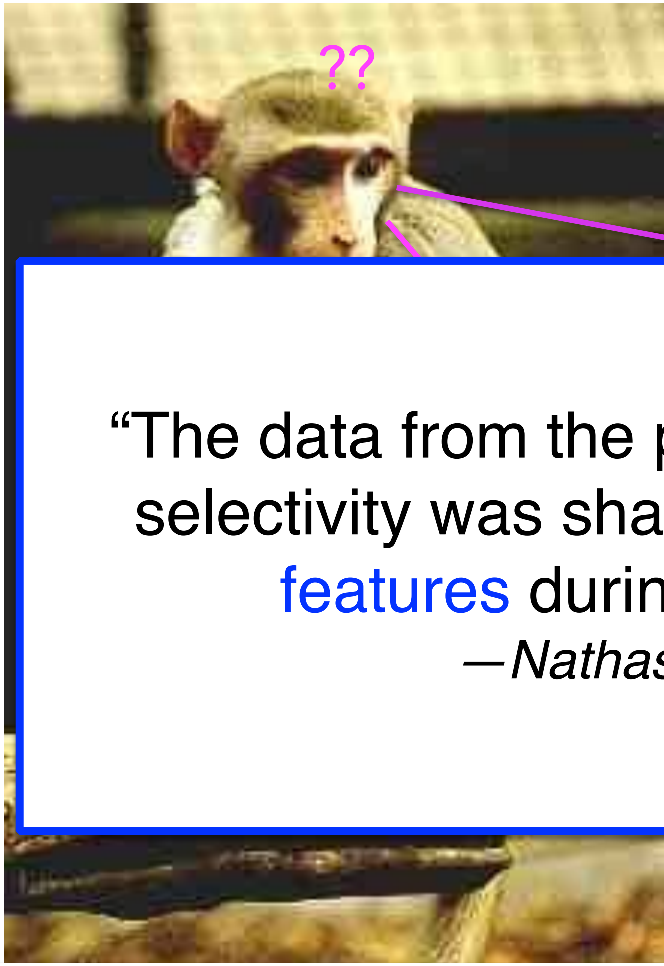
Outline

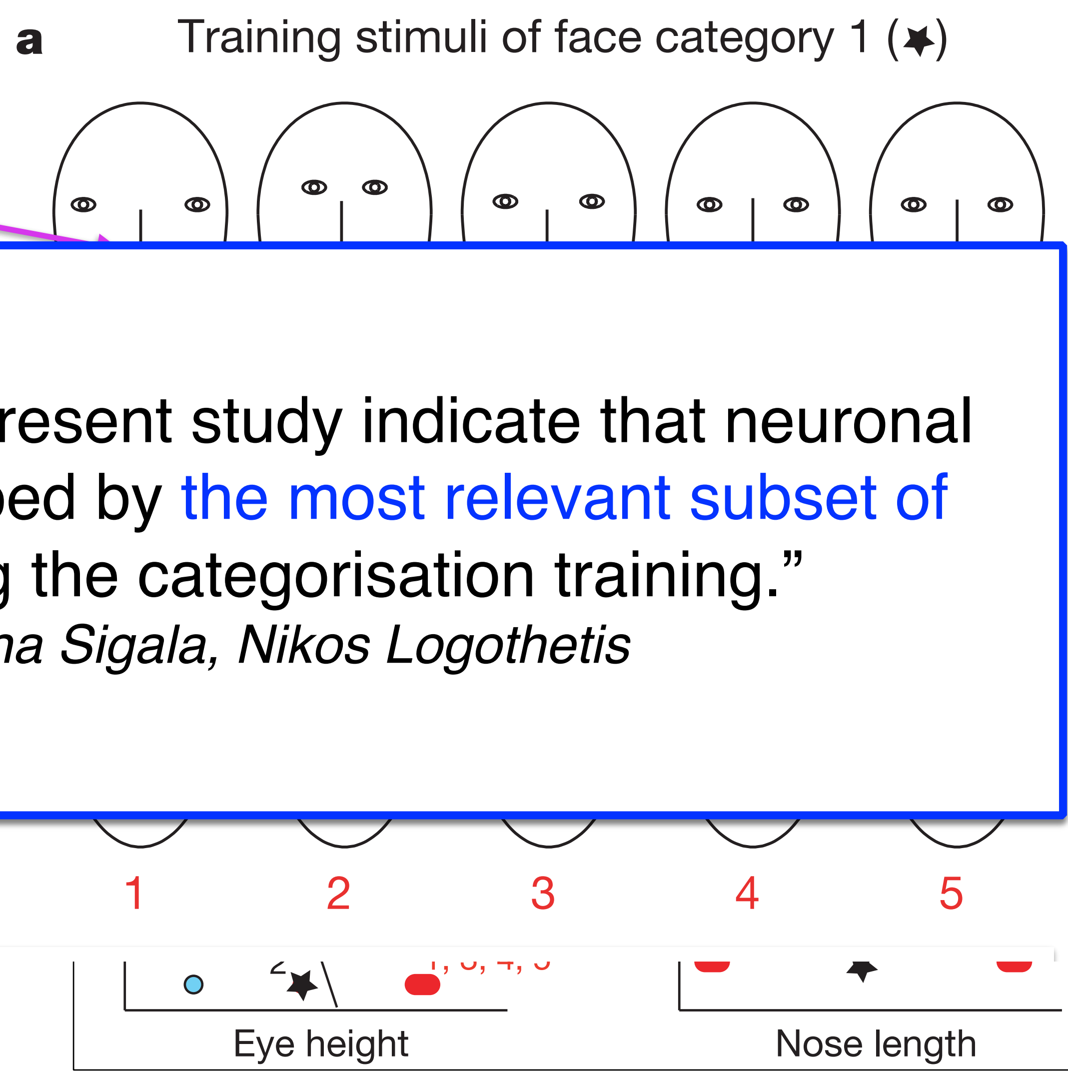
- Feature manipulation and feature selection
  - What is feature selection?
  - Why do feature selection?
  - Overall feature selection system
  - Feature selection bias
  - Wrapper, filter and embedded feature selection
  - Wrapper feature selection methods
  - Sequential search methods

Feature Manipulation

- A feature: X is a value (numerical/categorical) describing a characteristic of objects.
  - We often talk about feature vectors: multiple characteristics.
- Data transformations are mappings from the original input space to a new space.
- Feature manipulation is an umbrella term for input-space transformation or data transformation, including:
  1. feature ranking,
  2. dimensionality reduction
  3. feature (subset) selection
  4. feature construction, feature extraction, feature creation
  5. feature transformation

Feature Selection: Example from Biology



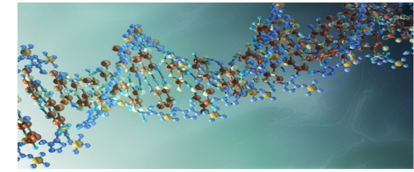


features)

"The data from the present study indicate that neuronal selectivity was shaped by the most relevant subset of features during the categorisation training."

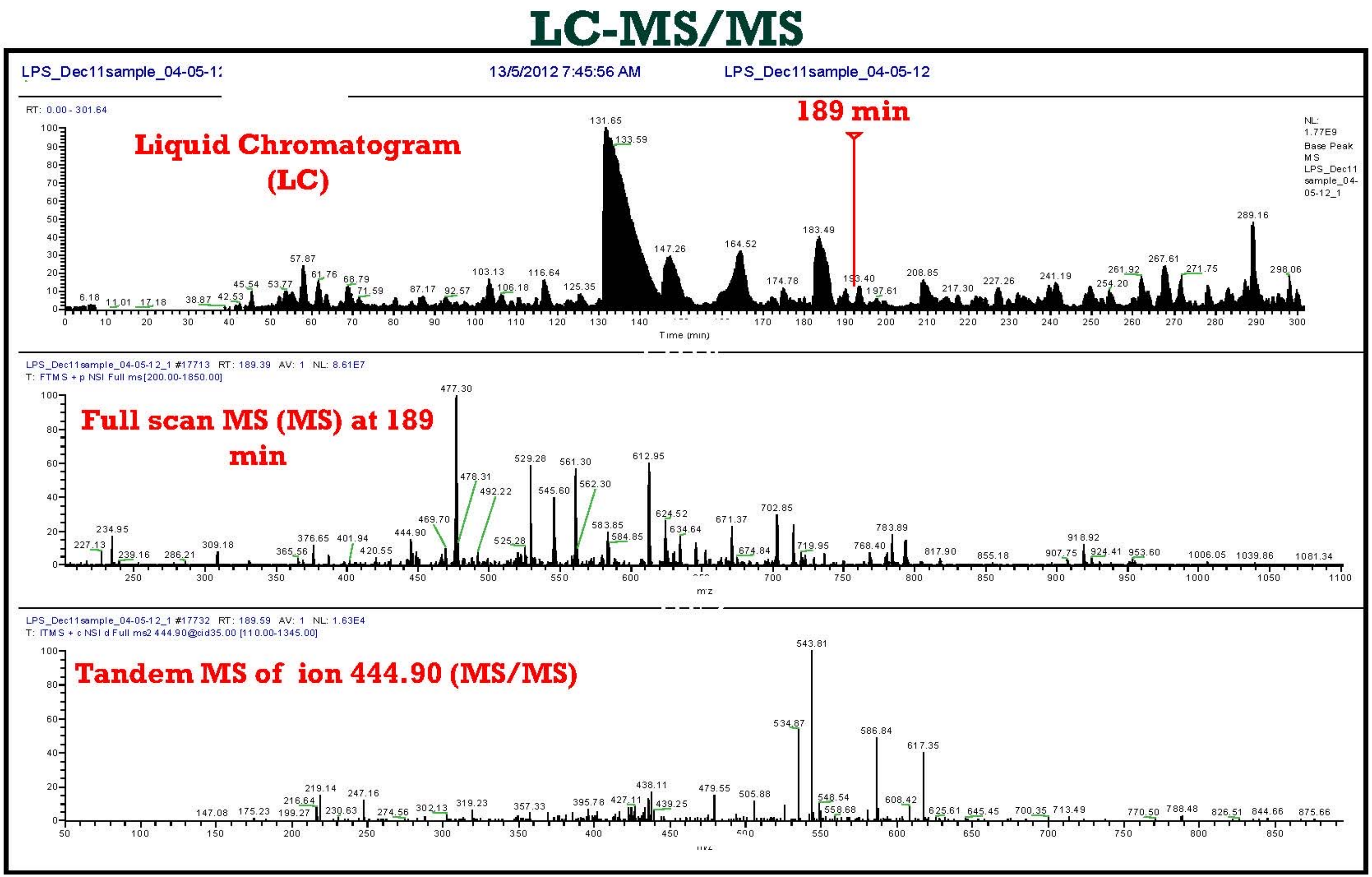
—*Nathasha Sigala, Nikos Logothetis*

[Acknowledgement: Nathasha Sigala, Nikos Logothetis: Visual categorization shapes feature selectivity in the primate visual cortex. *Nature* Vol. 415(2002)]



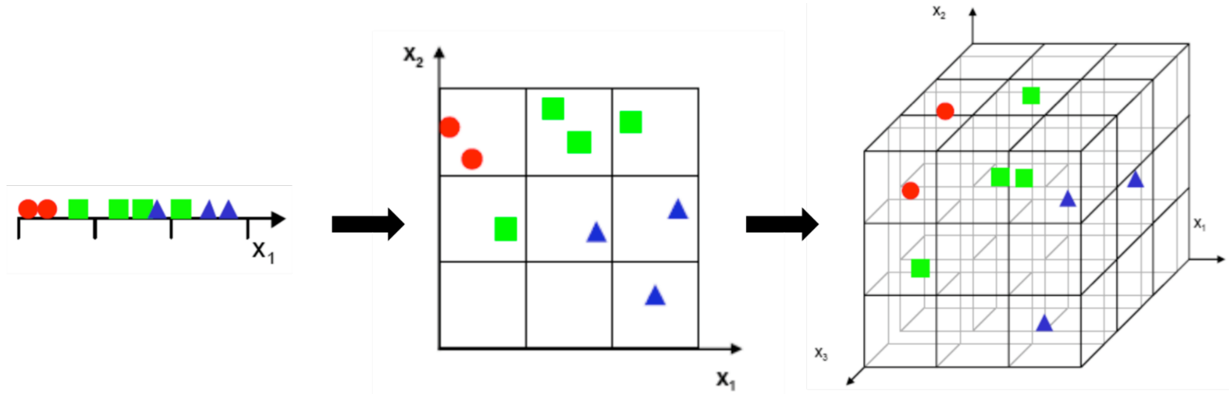
High-Dimensional Data

- Cancer Diagnosis



Why Do Feature Selection ?

- "Curse of dimensionality"
  - Large number of features: 100s, 1000s, even millions

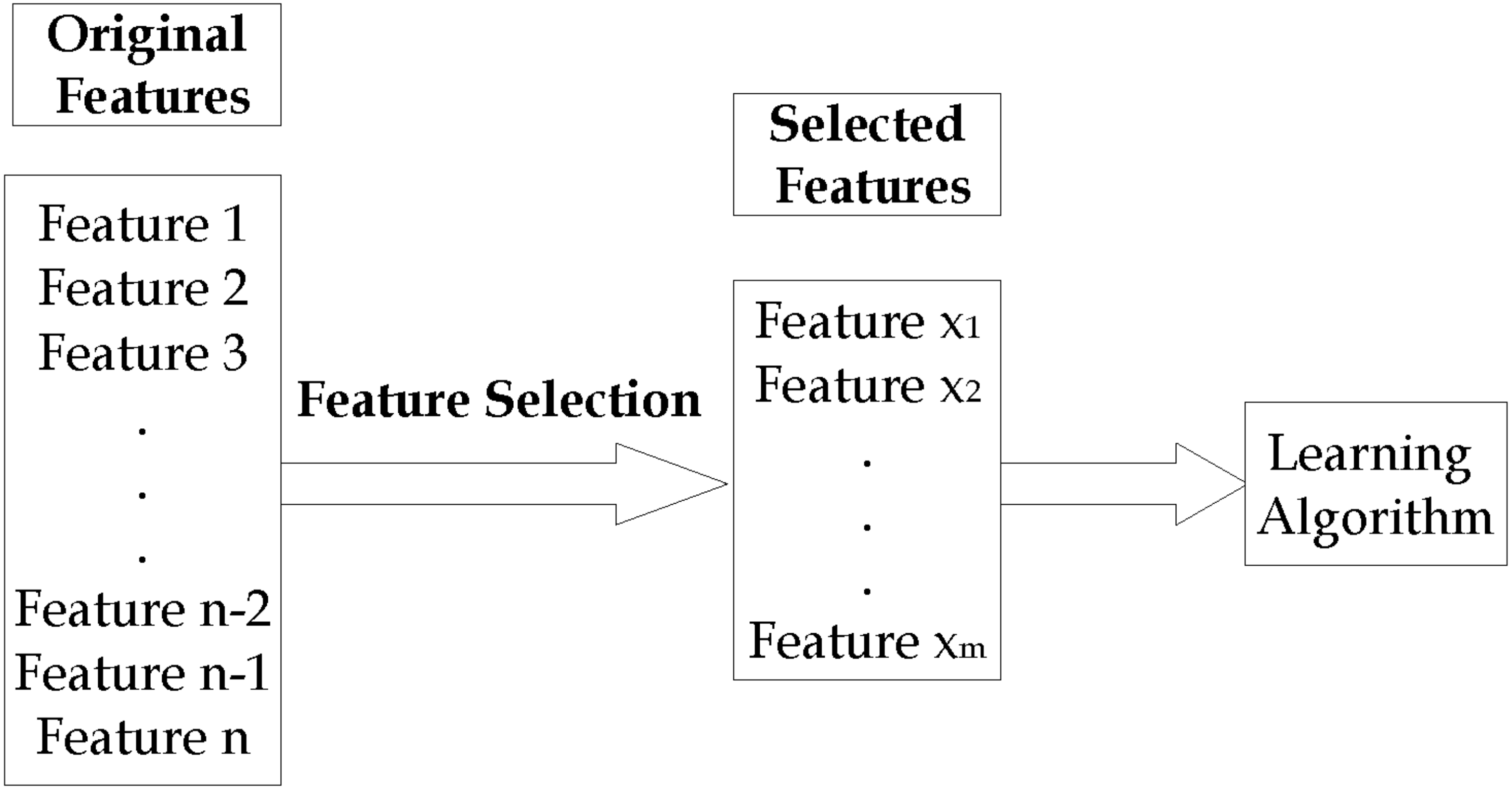


3 samples per region | 1 sample per region | 1/3 sample per region

  - Data density decreases exponentially with dimensionality ☹
- Not all features are useful (relevant)
- Redundant or irrelevant features may reduce the performance (e.g. classification accuracy). Can confuse many learning algorithms. How?
  - Naïve Bayes: $P(C \mid X_1, X_2, X_3) \sim P(C) * P(X_1 \mid C) * P(X_2 \mid C) * P(X_3 \mid C)$
- Costly: time, memory, and money

What is Feature Selection?

- Relevant vs irrelevant vs redundant features
- Feature selection
  - Select a small subset of **relevant** features from the original large set of features



Feature Selection Definitions

- **Classical**: to select *m* features from *n* original features, $m < n$, such that the value of a *criterion* function is optimised over all subsets of size *m*.
- **Idealised**: to find the *minimally* sized feature subset that is necessary and sufficient to describe the target concept.
- Improve classification accuracy/reduce complexity: improve classification performance *and/or* reduce model complexity.
- Approximating original class distribution: to select a subset of features such that the resulting class distribution, given only the selected features, is as close as possible to the original class distribution given by all the available features.

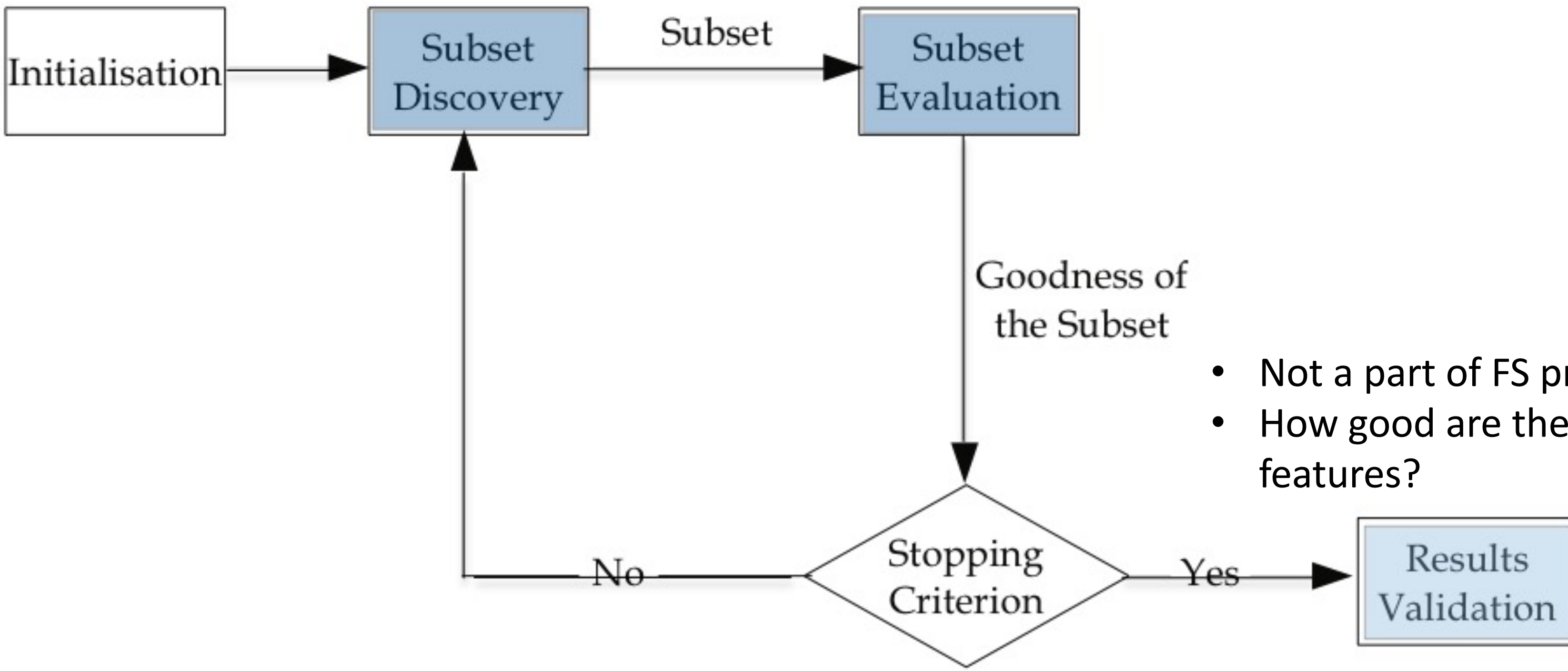
What can feature selection do ?

- Improve the (classification) performance
- Reduce the dimensionality (num of features)
- Simplify the learnt model
- Speed up the processing time
- Help visualisation and interpretation
- Reduce cost, e.g. save memory
- Can we achieve all objectives at the same time?
  - Multi-objective…

Feature Selection Process

Generate new feature subsets

Evaluate the goodness of the generated subsets



- Not a part of FS process
- How good are the selected features?

- Number of iterations
- Number of features
- ...

Challenges in Feature Selection

- Large search space: 2^N possible feature subsets
  - 1990: n < 20
  - 1998: n <= 50
  - 2007: n ≈ 100s
  - Now: 1000s, 1 000 000s
  - Big data ??

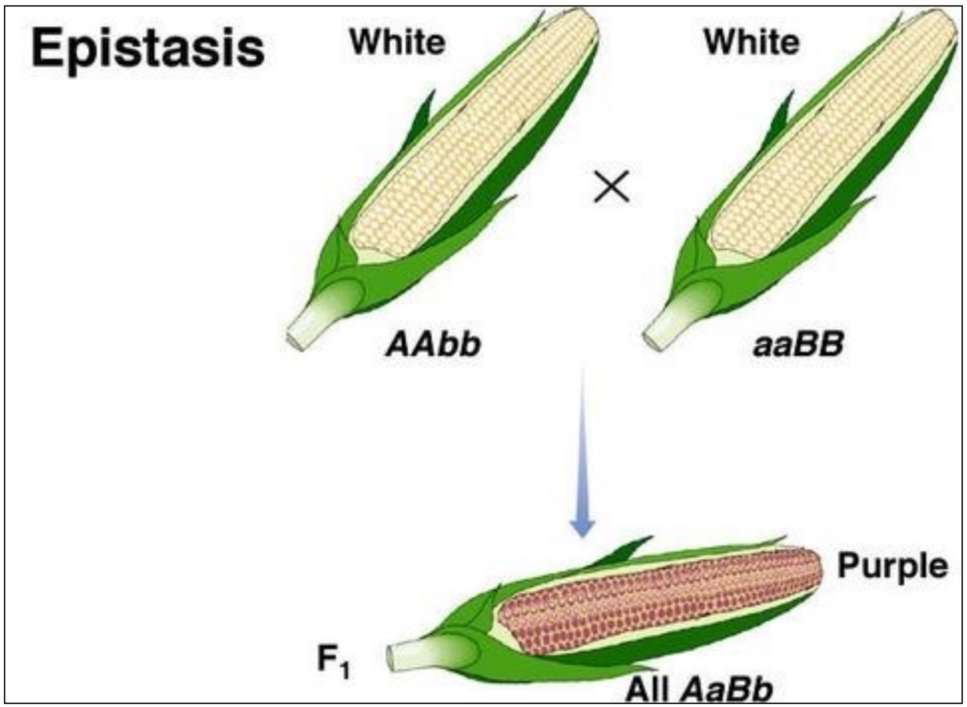
- *Feature interaction*
  - Relevant features may be (mutually) redundant
  - "Weakly relevant" features may become highly useful

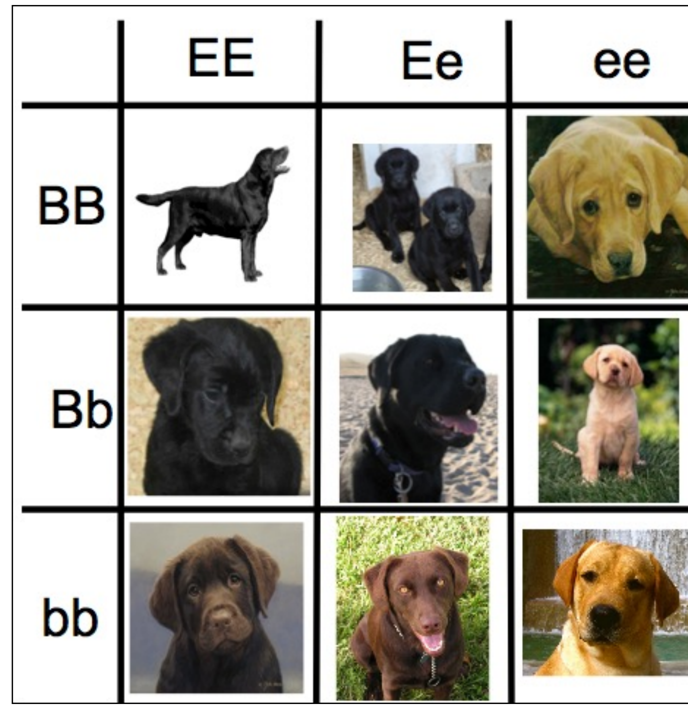
- Slow processing time, or even not possible:
  - 30 features -> 1,073,741,824 subsets -> 35 years (1 sec per subset)

- Multi-objective Problems

Feature Interactions

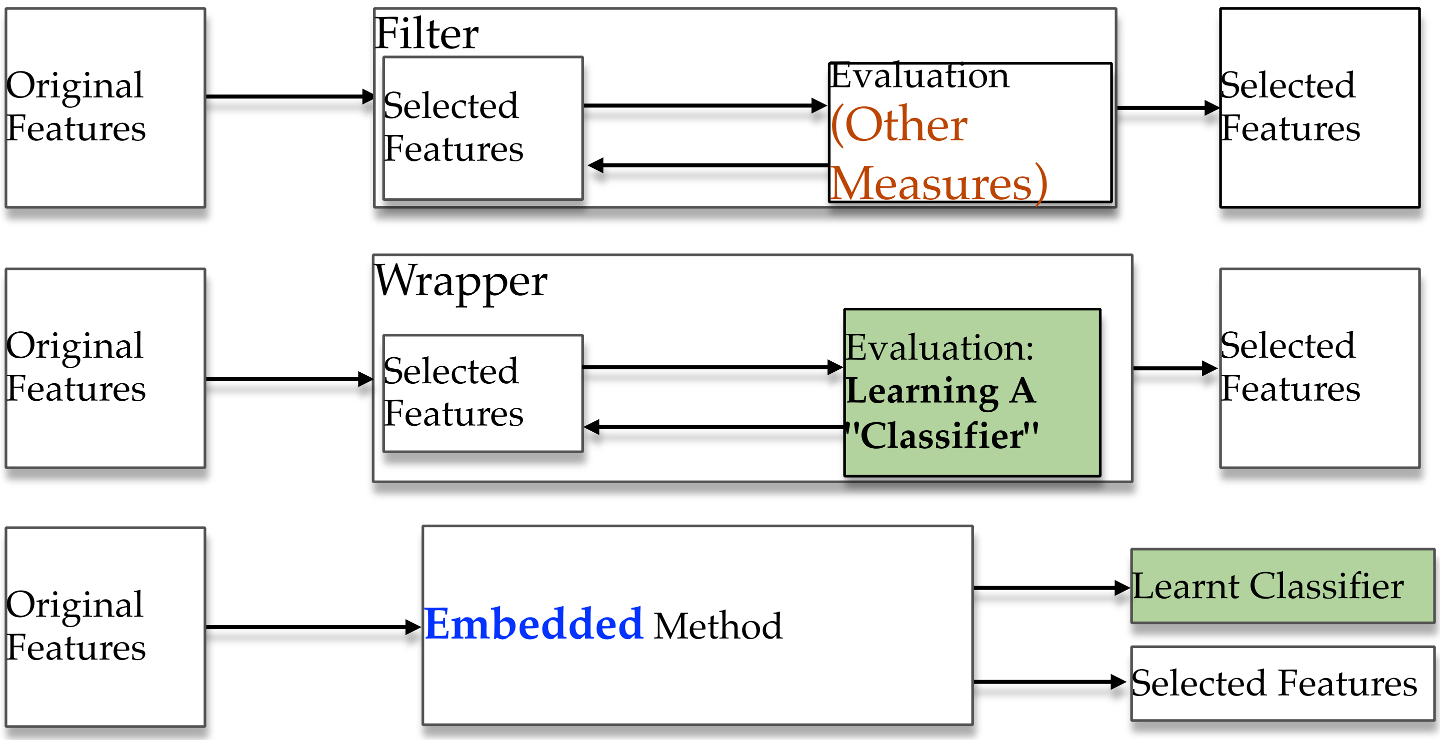
- Epistasis in biology: the appearance depends on the interactions between genes





Feature Selection Approaches

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

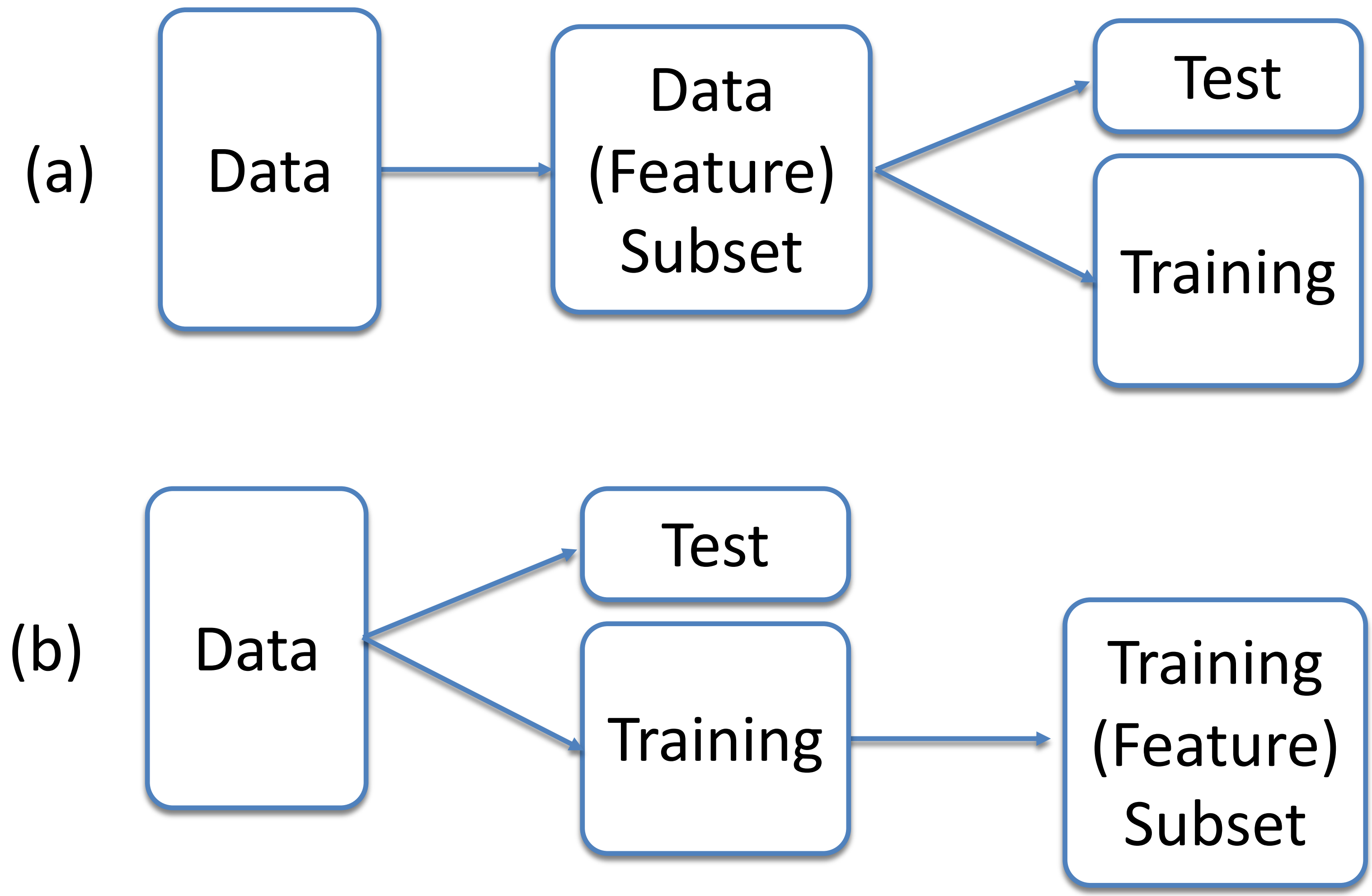


Feature Selection Approaches

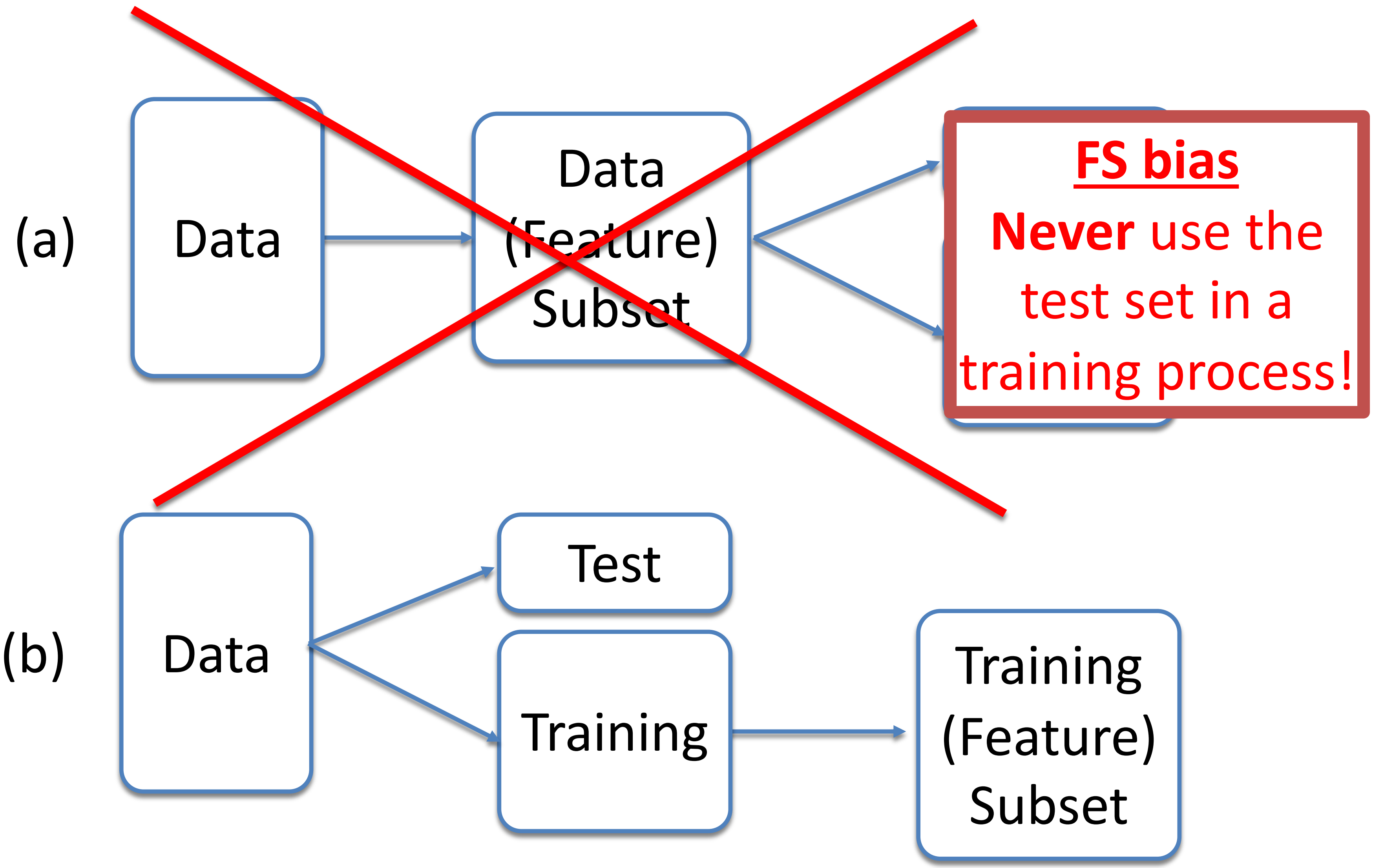
Generally:

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

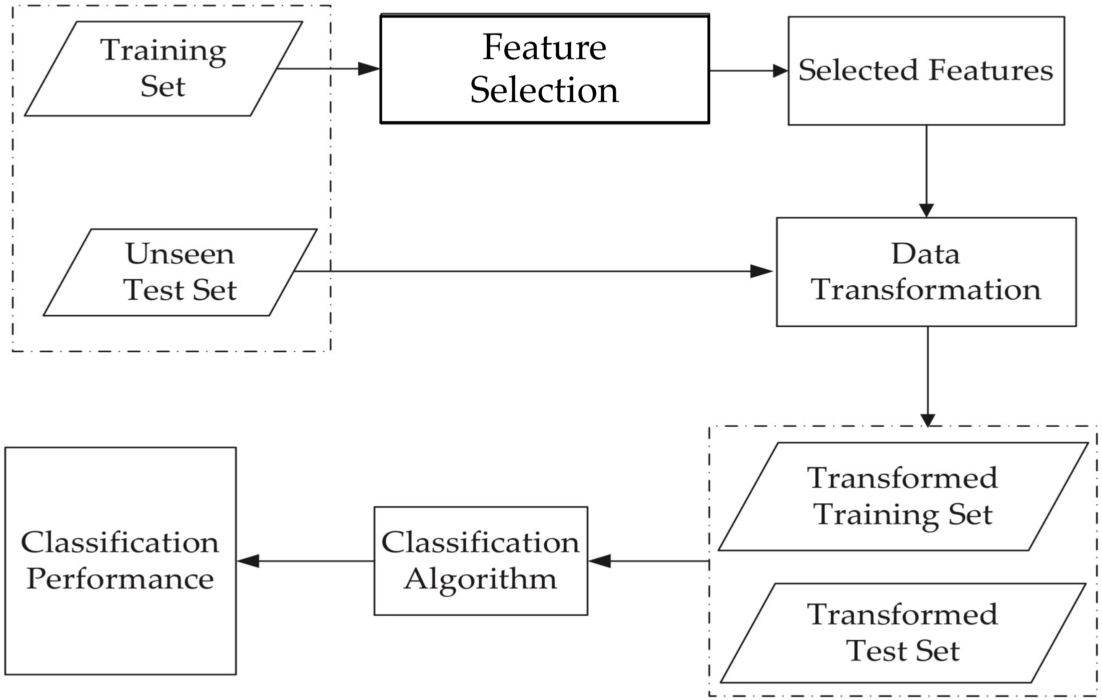
Any difference?



Any difference?



General FS/FC System: FS/FC Bias

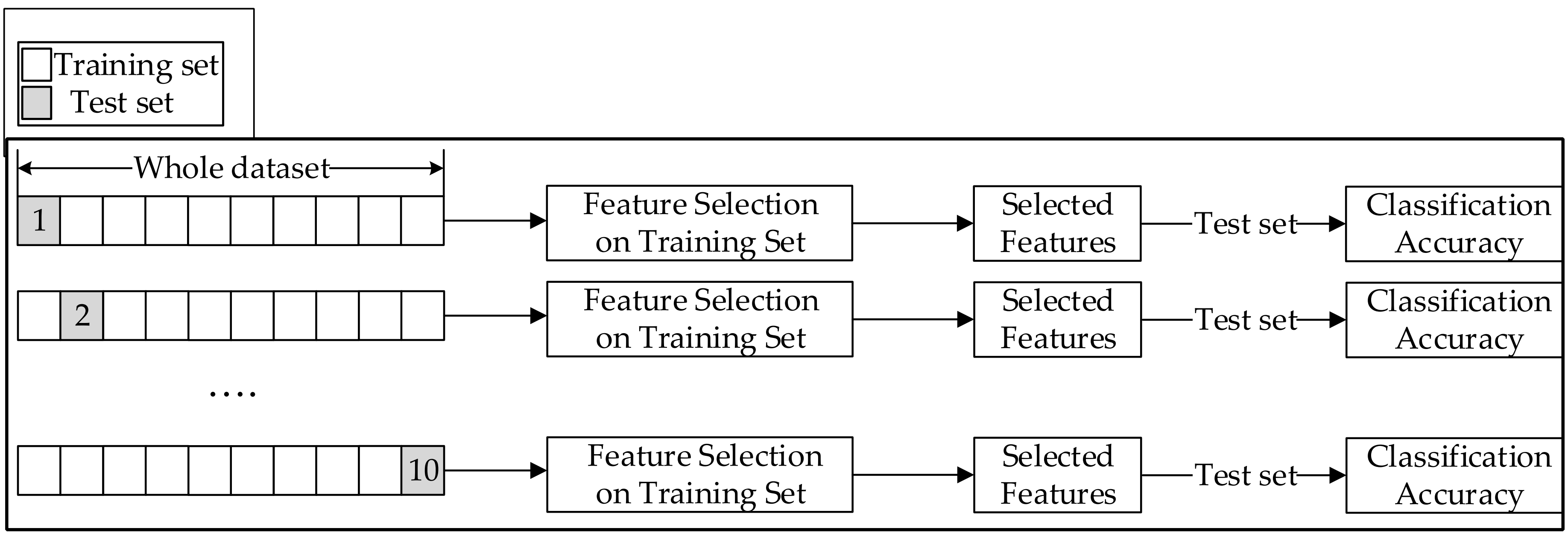


- If the whole dataset is used during FS process, the experiments(or evaluation) have ***feature selection bias***
- What if only a small number of instances available ?
  - In classification, we use k-fold cross validation
  - How can we use k-fold cross validation to evaluate a FS system?

Kohavi, Ron, and George H. John. "Wrappers for feature subset selection." Artificial intelligence 97.1-2 (1997): 273-324.

K-CV for FS/FC without Bias: Outer Loop

- k-fold cross validation (K-CV) in FS/FC to evaluate a FS/FC system without bias
- Use 10-CV for FS as an example
  - repeat FS 10 times
  - Use the average test accuracy as the final performance

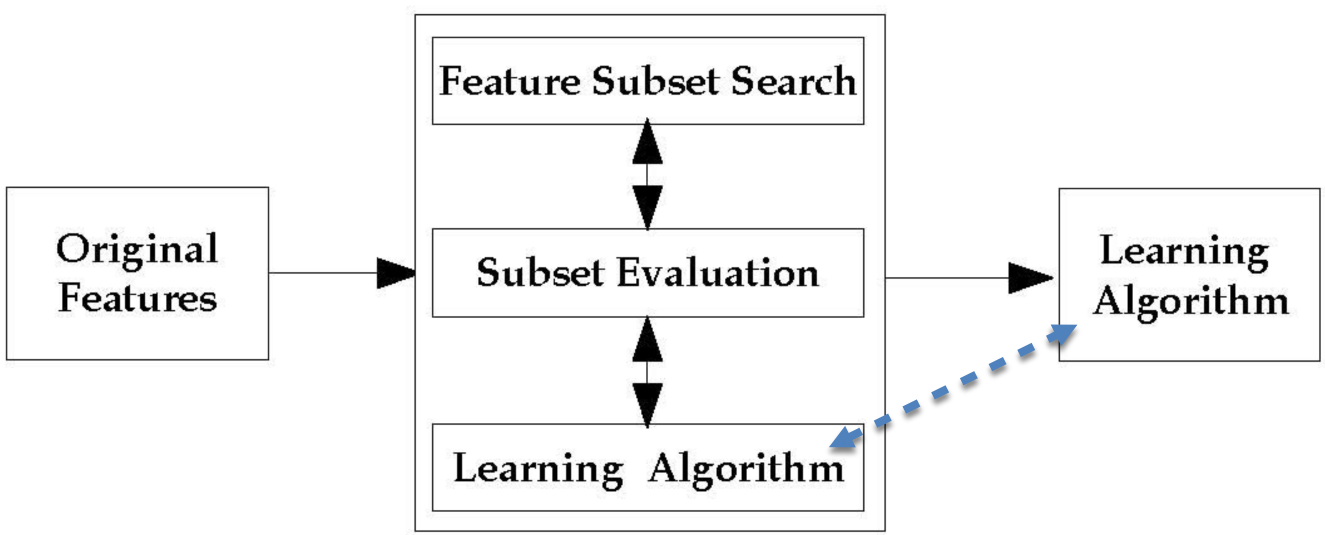


Kohavi, Ron, and George H. John. "Wrappers for feature subset selection." Artificial intelligence 97.1-2 (1997): 273-324.

WRAPPER FEATURE SELECTION

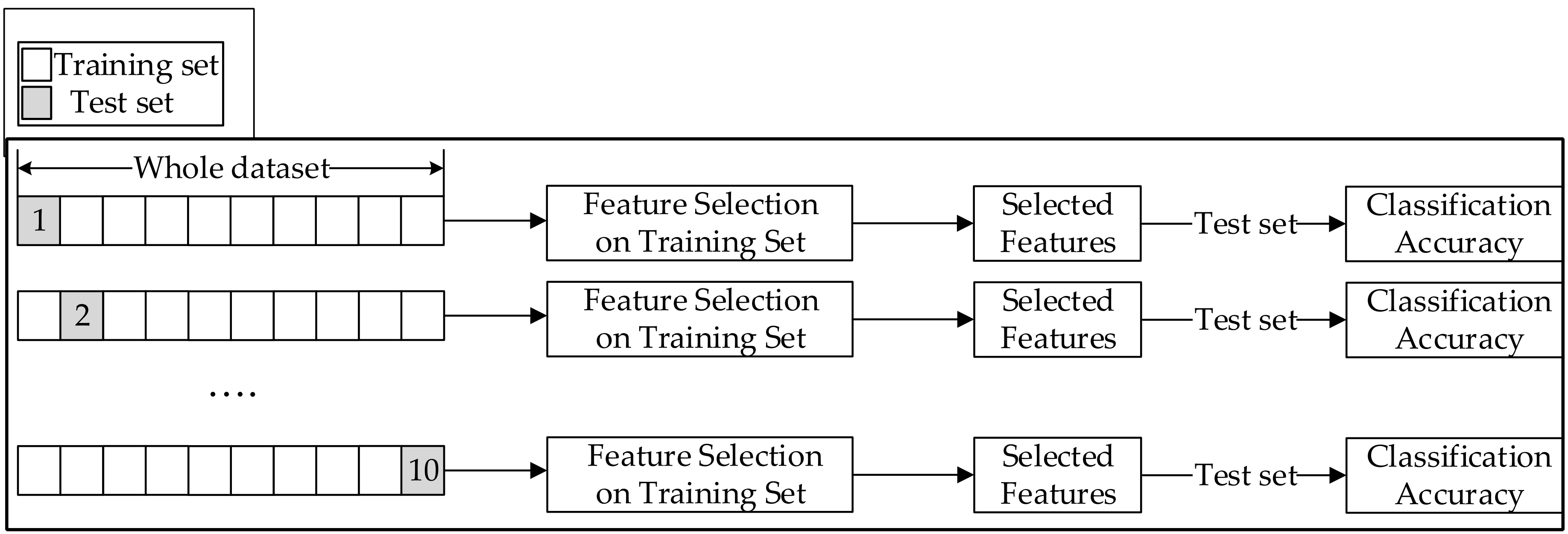
Wrapper Feature Selection

- A wrapper approach uses a learning algorithm for evaluation
- The goodness of a feature subset is (*partially*) measured by the learning performance (e.g. classification accuracy)
- Each evaluation involves training a learning algorithm
- Pros and cons:
  - Better results ✅
  - Computationally more expensive ❌
  - Less general to other classification algorithms ❌



K-CV for FS/FC without Bias: Outer Loop

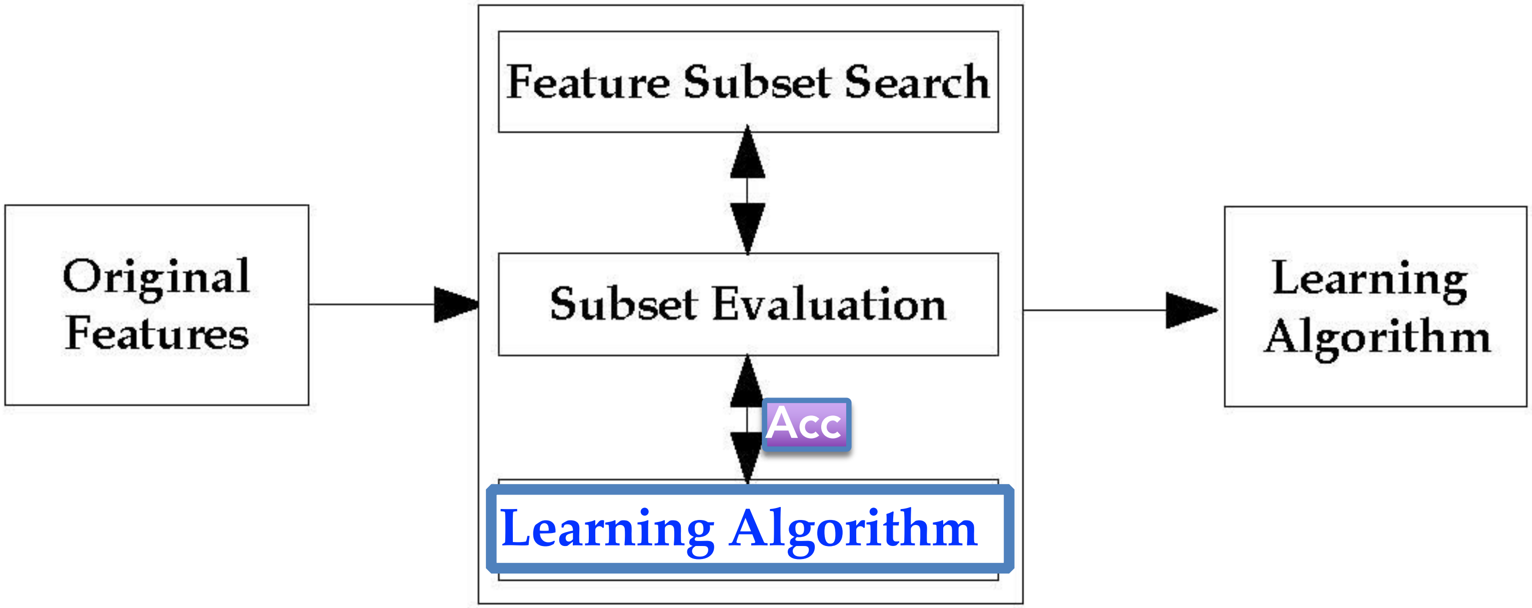
- k-fold cross validation (K-CV) in FS/FC to evaluate a FS/FC system without bias
- Use 10-CV for FS as an example
  - repeat FS 10 times
  - Use the average test accuracy as the final performance



Kohavi, Ron, and George H. John. "Wrappers for feature subset selection." Artificial intelligence 97.1-2 (1997): 273-324.

K-CV for *Wrapper* FS/FC without Bias

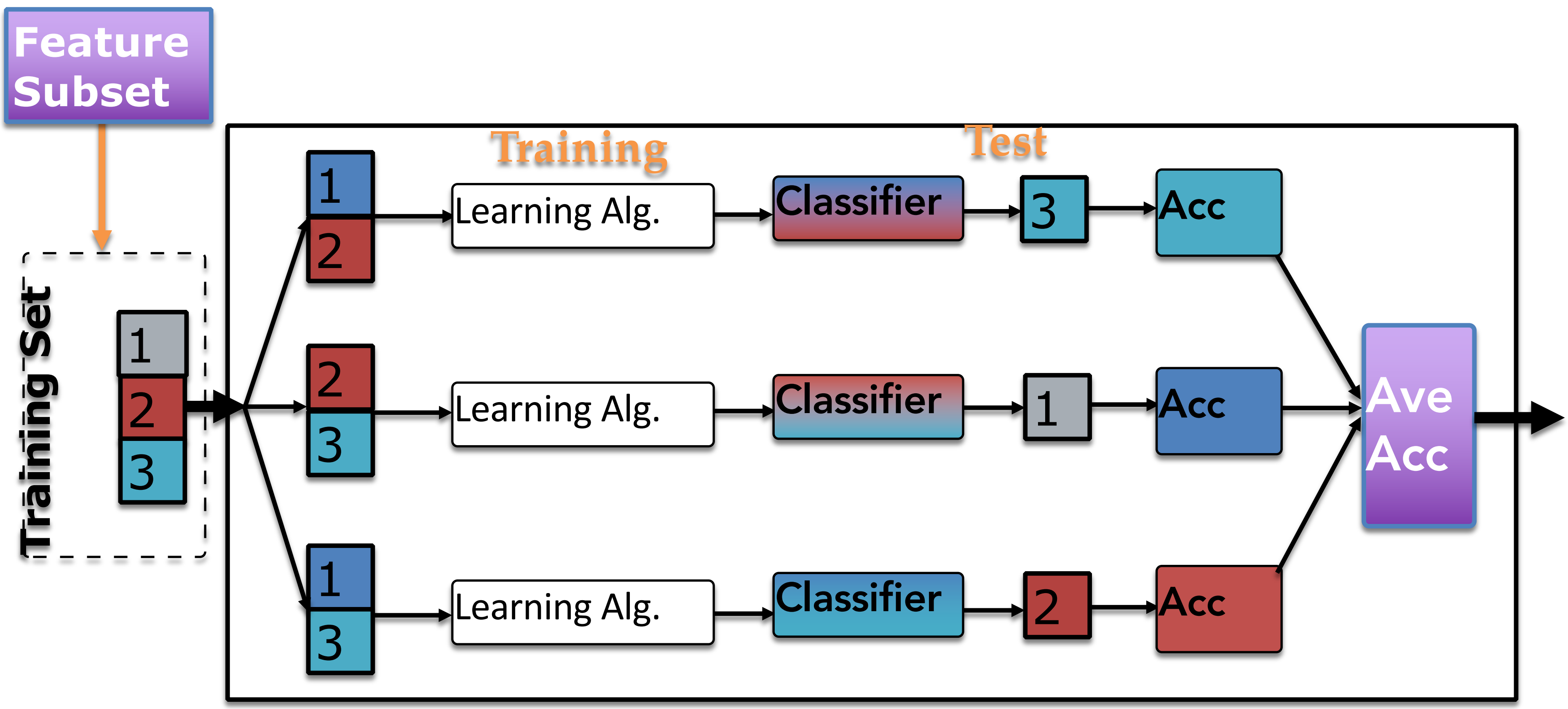
- *Wrapper*: each evaluation involves a classification training and testing process: sub-training and sub-test sets
- How to use K-CV to evaluate a wrapper FS/FC system ?
- ***Outer*** loop and ***inner*** loop



Kohavi, Ron, and George H. John. "Wrappers for feature subset selection." Artificial intelligence 97.1-2 (1997): 273-324.

K-CV in Each Evaluation — Inner Loop

- 3-CV as an *inner loop* to evaluate each feature subset
- In **each** evaluation to get Acc



Kohavi, Ron, and George H. John. "Wrappers for feature subset selection." Artificial intelligence 97.1-2 (1997): 273-324.

SEQUENTIAL SEARCH

Sequential Forward selection (SFS) (heuristic, greedy search)

1. The best single feature is selected (by some criteria)

2. Pairs of features are formed using the selected feature and each remaining feature. The best pair is selected.

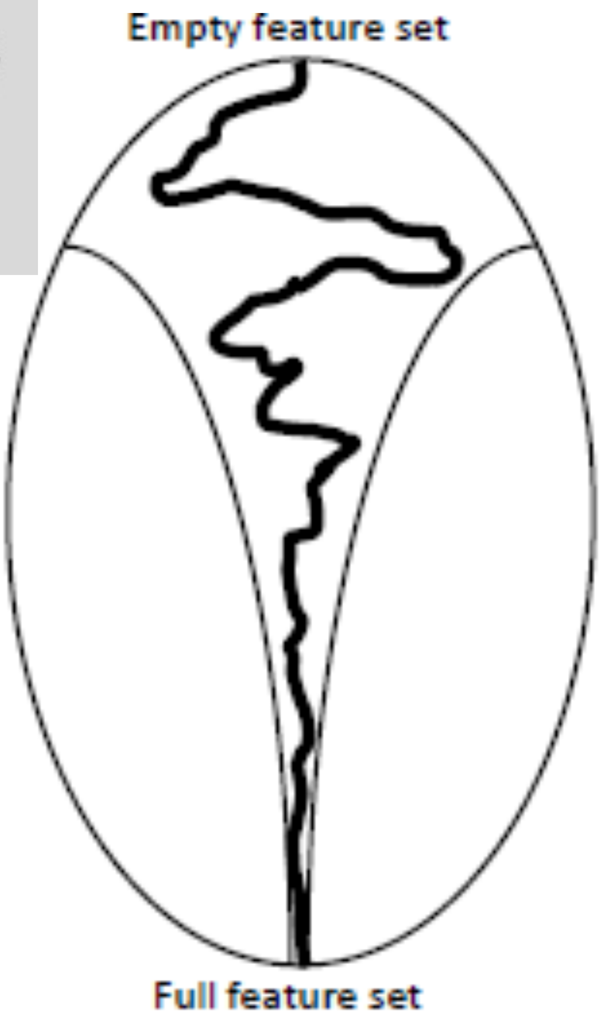
3. Triplets of features are formed using the selected features and each remaining feature. The best triplet is selected.

4. This procedure continues until a predefined number of features are selected or criterion value not improved.

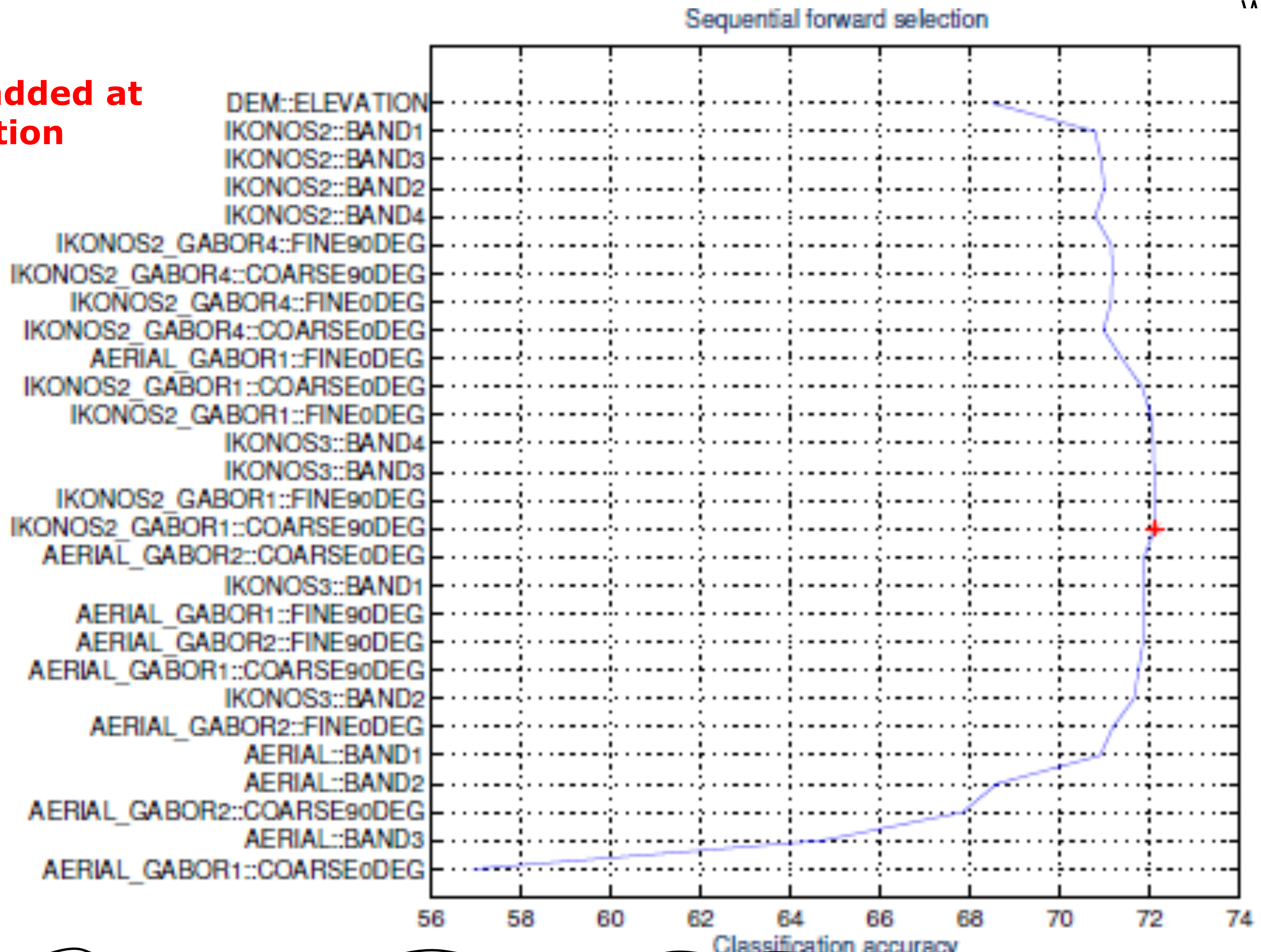
Sequential Forward selection (SFS) (heuristic, greedy search)

1. Start with the empty set $Y_0 = \{\emptyset\}$
2. Select the next best feature $x^+ = \underset{x \notin Y_k}{\arg\max} J(Y_k + x)$
3. Update $Y_{k+1} = Y_k + x^+;\ k = k + 1$
4. Go to 2

SFS performs best when the optimal subset is small.



Features added at each iteration



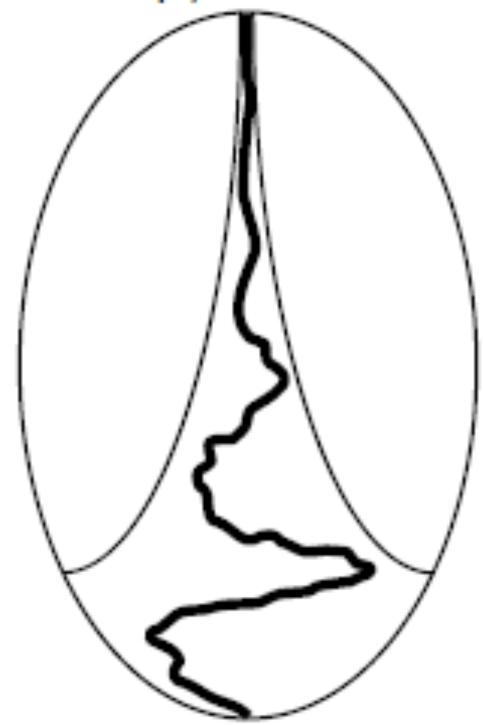
SFS for classification of a satellite image (28 features)
x-axis: classification accuracy (%)
y-axis: shows the features added at each iteration
The highest accuracy value is shown with a +

Sequential Backward selection (SBS) (heuristic search)

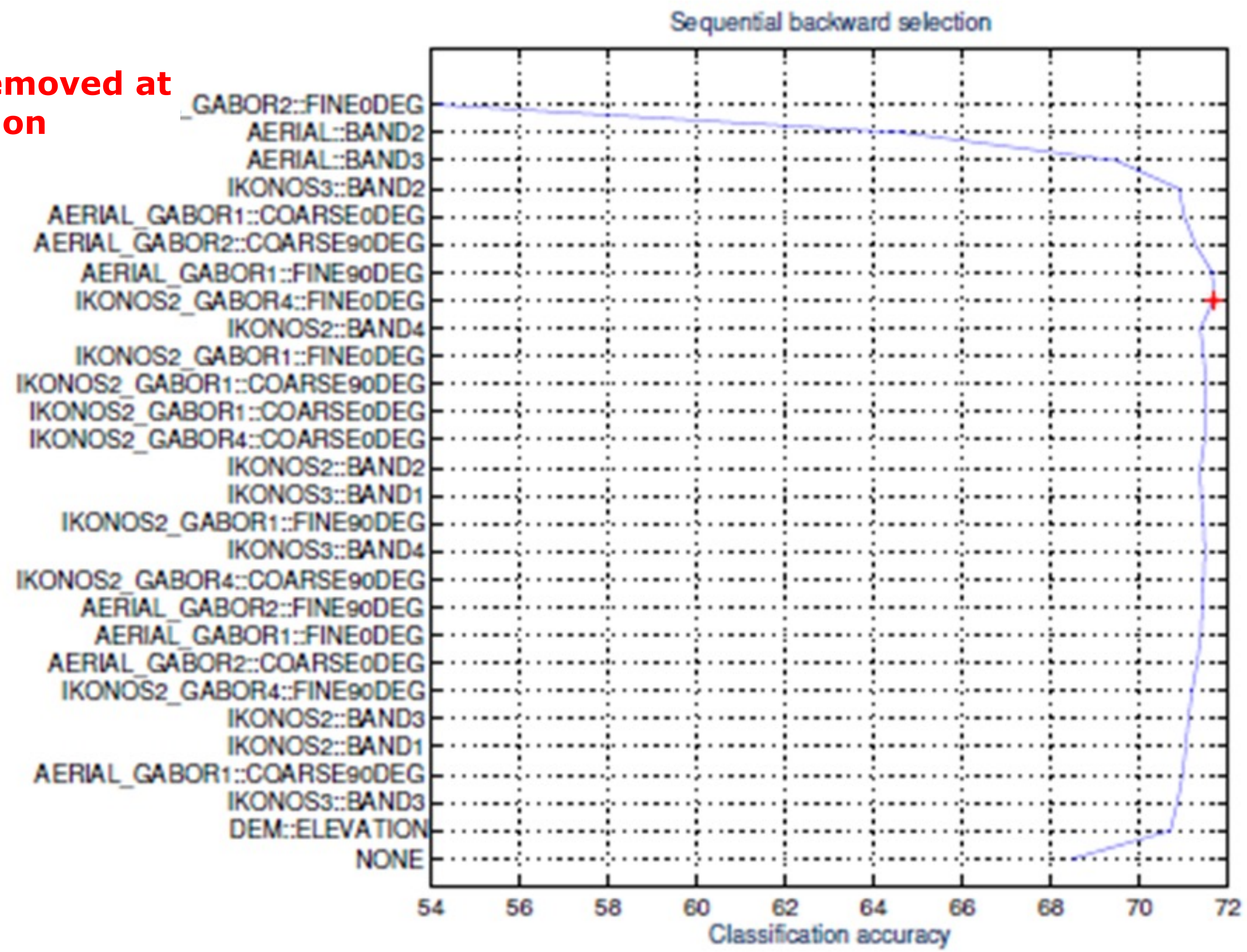
- Opposite to SFS: start with all features selected
- Iteratively remove the worst feature from the feature subset
- Requires computing criterion value for n-1 subsets at the 1st iteration…

1. Start with the full set $Y_0 = X$
2. Remove the worst feature $x^- = \arg\max_{x \in Y_k} J(Y_k - x)$
3. Update $Y_{k+1} = Y_k - x^-;\ k = k + 1$
4. Go to 2

SFS performs best when the optimal subset is large.



Features removed at each iteration

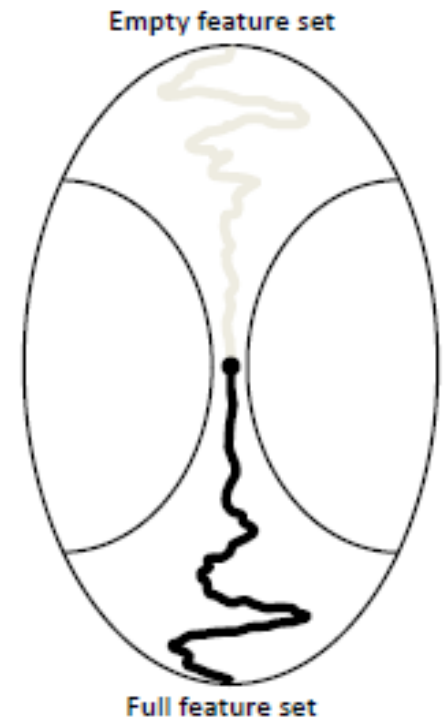


Bidirectional Search (BDS)

BDS applies SFS and SBS simultaneously:

- SFS starts from the empty set
- SBS starts from the full set

- To guarantee that SFS and SBS converge to the same solution:
  - Features already selected by SFS are not removed by SBS.
  - Features already removed by SBS are not added by SFS.



1. Start SFS with $Y_F = \{\emptyset\}$
2. Start SBS with $Y_B = X$
3. Select the best feature

$$x^+ = \underset{\substack{x \notin Y_{F_k} \\ x \in \mathrm{F}_{B_k}}}{\arg\max} J\left(Y_{F_k} + x\right)$$

$$Y_{F_{k+1}} = Y_{F_k} + x^+$$

4. Remove the worst feature

$$x^- = \underset{\substack{x \in Y_{B_k} \\ x \notin Y_{F_{k+1}}}}{\arg\max} J\left(Y_{B_k} - x\right)$$

$$Y_{B_{k+1}} = Y_{B_k} - x^-; \; k = k + 1$$

5. Go to 2

Limitations of SFS and SBS

Nesting problem

- SFS cannot remove features that become unuseful after the addition of other features
- SBS cannot re-evaluate the usefulness of a feature after it has been discarded

- Some generalisations of SFS and SBS:
  - "Plus-L, minus-R" selection (LRS)
  - Sequential floating forward/backward selection (SFFS and SFBS)

"Plus-L, minus-R" Selection (LRS)

A generalisation of SFS and SBS

If L>R, LRS starts from the empty set and:

- Repeatedly add L features
- Repeatedly remove R features

If L<R, LRS starts from the full set and:

- Repeatedly removes R features
- Repeatedly add L features

Its main limitation is the lack of theory to choose the optimal values of L and R

1. If L>R then $Y_0 = \{\emptyset\}$
   else $Y_0 = X$; go to step 3
2. Repeat L times
   $$x^+ = \arg\max_{x \notin Y_k} J(Y_k + x)$$
   $$Y_{k+1} = Y_k + x^+;\ k = k + 1$$
3. Repeat R times
   $$x^- = \arg\max_{x \in Y_k} J(Y_k - x)$$
   $$Y_{k+1} = Y_k - x^-;\ k = k + 1$$
4. Go to 2

SFFS and SFBS

- An extension to LRS:
  - Rather than fixing the values of L and R, floating methods determine these values from the data
  - The dimensionality of the subset during the search can be thought to be "floating" up and down

- Two floating methods:
  - Sequential floating forward selection (SFFS)
  - Sequential floating backward selection (SFBS)

P. Pudil, J. Novovicova, J. Kittler, Floating search methods in feature selection, Pattern Recognition Lett. 15 (1994) 1119–1125.

Sequential floating forward selection (SFFS)

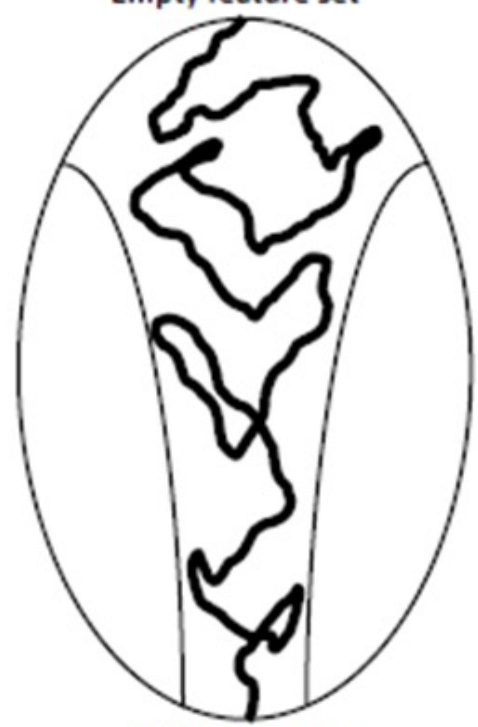
- Sequential floating forward selection starts from the empty set.

1. $Y = \{\emptyset\}$
2. Select the best feature
$$x^+ = \arg\max_{x \notin Y_k} J(Y_k + x)$$
$$Y_k = Y_k + x^+; k = k + 1$$
3. Select the worst feature*
$$x^- = \arg\max_{x \in Y_k} J(Y_k - x)$$
4. If $J(Y_k - x^-) > J(Y_k)$ then
$$Y_{k+1} = Y_k - x^-;\ k = k + 1$$
Go to step 3

   Else

   Go to step 2

- After each forward step, SFFS performs backward steps as long as the objective function increases.



Sequential floating backward selection (SFBS)

- Sequential floating backward selection (SFBS) starts from the full set.
- Perform backward selection:
  - After each backward step, SFBS performs forward steps as long as the objective function increases.

Reading list

- Guyon, Isabelle, and André Elisseeff. "An introduction to variable and feature selection." Journal of machine learning research 3.Mar (2003): 1157-1182.
- Kohavi, Ron, and George H. John. "Wrappers for feature subset selection." Artificial intelligence 97.1-2 (1997): 273-324.
- Tang, Jiliang, Salem Alelyani, and Huan Liu. "Feature selection for classification: A review." Data classification: Algorithms and applications (2014): 37.
- Xue, Bing, et al. "A survey on evolutionary computation approaches to feature selection." IEEE Transactions on Evolutionary Computation 20.4 (2015): 606-626.