Data Science for Business

Feature Selection and Basic Feature Engineering

Asst. Prof. Teerapong Leelanupab (Ph.D.)
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang (KMITL)

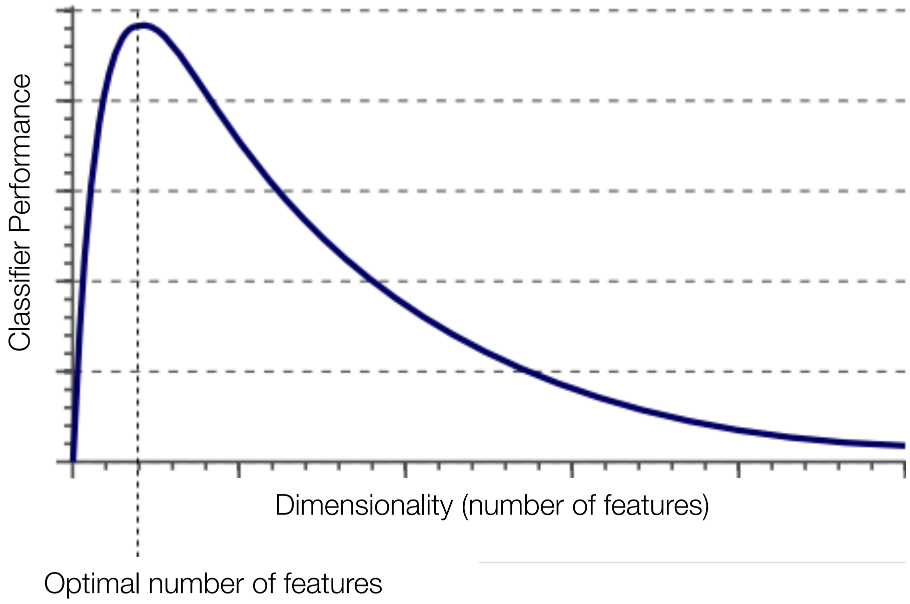
Week 11

Overview

- The Curse of Dimensionality
- Dimension Reduction
- Feature Transformation v Selection
- Feature Selection in Supervised Learning
    - Filter approaches
    - Wrapper approaches
    - Embedded feature selection
    - Overfitting in feature selection

Curse of Dimensionality

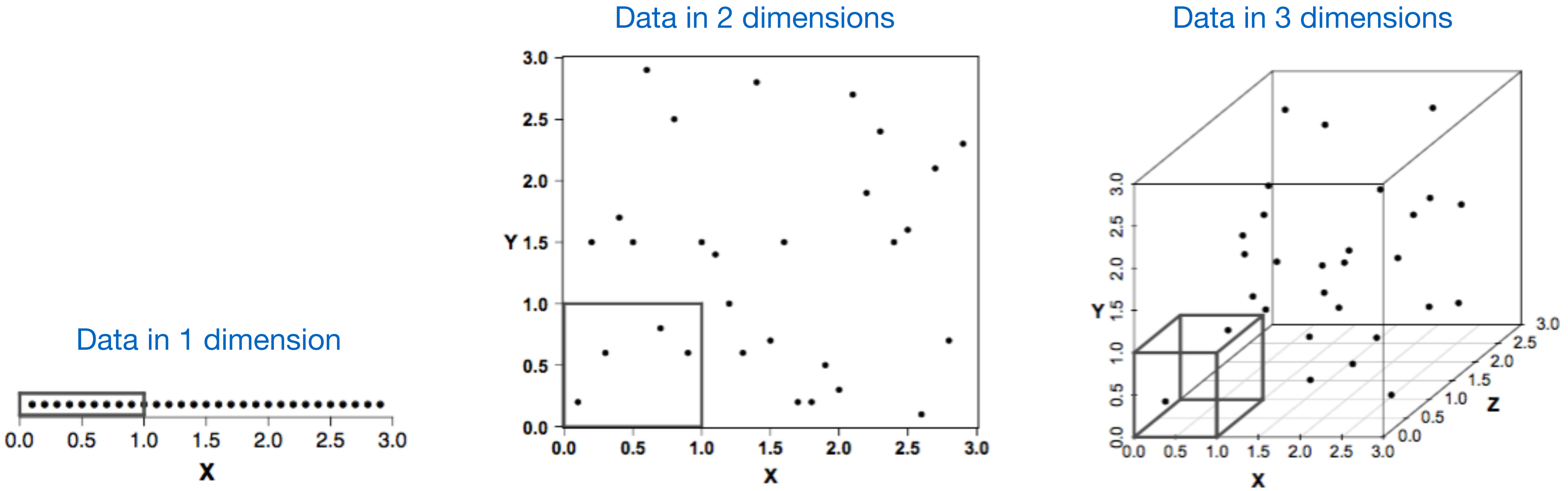
- Intuitively adding more features to a dataset should provide more information about each example, making prediction easier.
- In reality, we often reach a point where adding more features no longer helps, or can even reduce predictive power.



- Curse of dimensionality: refers to how many learning algorithms can perform poorly on high-dimensional data (Bellman, 1961).
- In practice, this means that, for a given number of examples, there is a maximum number of features beyond which the performance of a classifier will degrade rather than improve.

Curse of Dimensionality

- To build a model from data, the number of examples required per feature increases exponentially with number of features.
- High-dimensional spaces tend to be very *sparse*, so every point is equally far away from virtually every other point, and distances tend to be uninformative.



Data in 1 dimension

Data in 2 dimensions

Data in 3 dimensions

➡ The more dimensions you have, the more similar examples appear to one another.

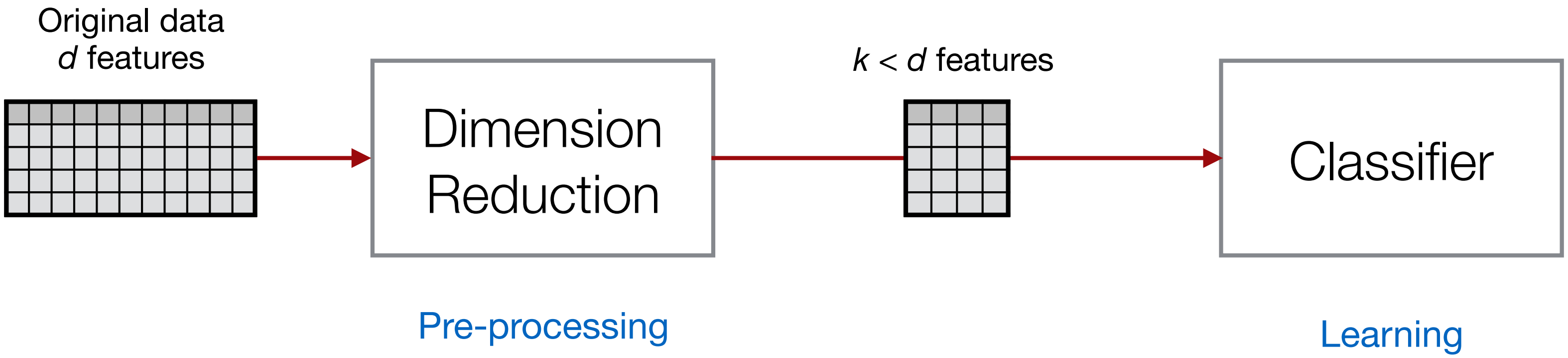
Dimension Reduction

- There are often other reasons why we might want to reduce the number of dimensions used to represent data:
  - **Computational cost**: For many algorithms, having a large number of features can significantly increase running times and memory usage.
  - **Financial cost**: In certain domains (e.g. clinical medicine, manufacturing), running experiments to generate data can be expensive. In such cases, we only want to generate the minimum number of features required for the task.
  - **Interpretability**: A feature set which is more compact can help to give a better understanding of the underlying process that generated the data.

➡ Understanding which features of our data are informative and which are not is an important knowledge discovery task.

Dimension Reduction

- We often try to beat the curse of dimensionality by applying pre-processing techniques to reduce the number of features.
- We then build a classification model on the smaller feature set.
- In many (but not all) cases, the additional information that is lost by removing some features is (more than) compensated by higher classifier accuracy in the lower dimensional space.



Feature Transformation v Selection

- There are two general strategies for dimension reduction:

  Feature Transformation (Feature Extraction)

  - Transforms the original features of a dataset to a completely new, smaller, more compact feature set, while retaining as much information as possible.
    e.g. Principal Components Analysis (PCA), Linear Discriminant Analysis (LDA)

  Feature Selection

  - Tries to find a minimum subset of the original features that optimises one or more criteria, rather than producing an entirely new set of dimensions for the data.

    e.g. Information Gain filter, Wrapper with sequential forward selection

Feature Selection in Classification

- Feature Subset Selection: Find the best subset of all available features, which contains the least number of dimensions that most contribute to accuracy. Discard the remaining, unimportant dimensions.

- Why select subset of original features?

  1. Building a better classifier - Redundant or noisy features can damage accuracy.
  2. Knowledge discovery - Identifying useful features helps us learn about the domain.
  3. Features expensive to obtain - Test a large number of features, select a few for the final system (e.g. sensors, manufacturing).
  4. Interpretability - Selected features still have meaning. We can extract meaningful rules from the classifier.

Feature Selection Strategies

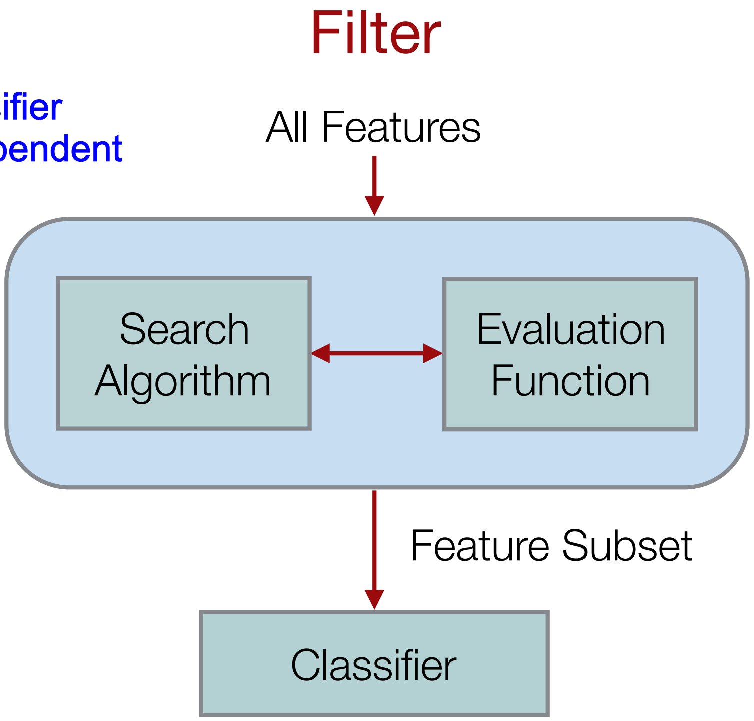
- Finding the optimal feature subset for a given dataset is difficult.
- Brute force evaluation of all feature subsets involves $\binom{d}{k}$ combinations if k is fixed, or 2^d combinations if not fixed.

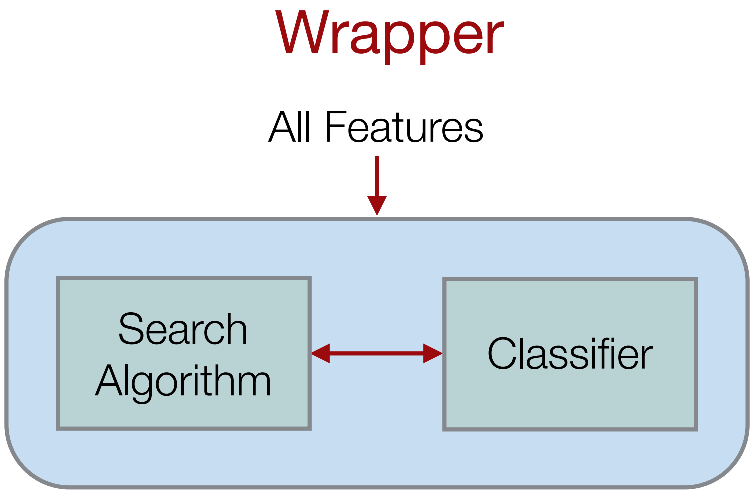
2 มาจาก เอากับไม่เอา

- Two broad strategies for feature selection:

diffrerent

classifier independent





นำ classifier มาเป็นเกณฑ์ในการ select feature

เลือก feature โดยเป็นอิสระจาก classifier
ข้อเสียที่ว่า feature ที่ดีอาจจะไม่สอดคล้องกับตัว classifier

Filters

- Pre-processing step that ranks and “filters” features independently of the choice of classifier that will be subsequently applied.
- **Evaluation function:** How does a filter algorithm score different feature subsets to produce an overall ranking?
- Generally score the predictiveness of the features.
    - Information theoretic analysis

        e.g. Information Gain, Breiman’s Gini index
    - Statistical tests

        e.g. Chi-square statistic
    - Relief algorithm

        Filter for binary classification, based on the nearest-neighbour classification algorithm (Kira & Rendell, 1992).

ขึ้นอยู่กับ model ที่เราอยากจะเลือกใช้ด้วย เช่น decision tree ต้องการอะไร ก็จะไปใช้ feature selection อันไหน

Information Gain Filter

- Given a set of training examples S, where p is the proportion of positive examples, q is the proportion of negative examples.

Entropy $H(S) = -p\log_2(p) - q\log_2(q)$

- A feature f that is predictive of a class will give significant Information Gain (i.e. a reduction in uncertainty):

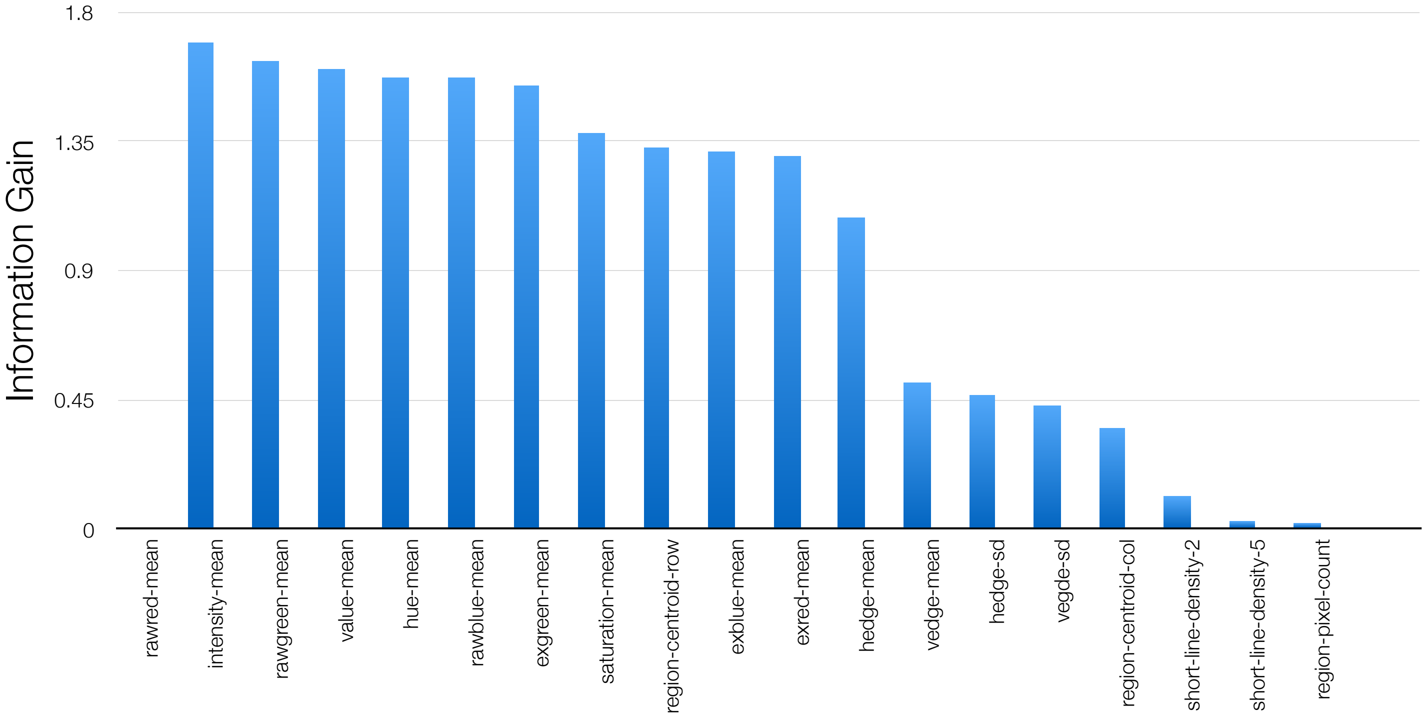
$$IG(S, f) = H(S) - \sum_{v \in values(f)} \frac{|S_v|}{S} H(S_v)$$

แยกข้อมูลได้เยอะ information gain ก็จะสูง (ใน dcs tree)

- **IG Filter approach**:
  1. Score all features based on their Information Gain (IG).
  2. Rank features based on their scores.
  3. Select a subset of the top ranked features to use for classification.

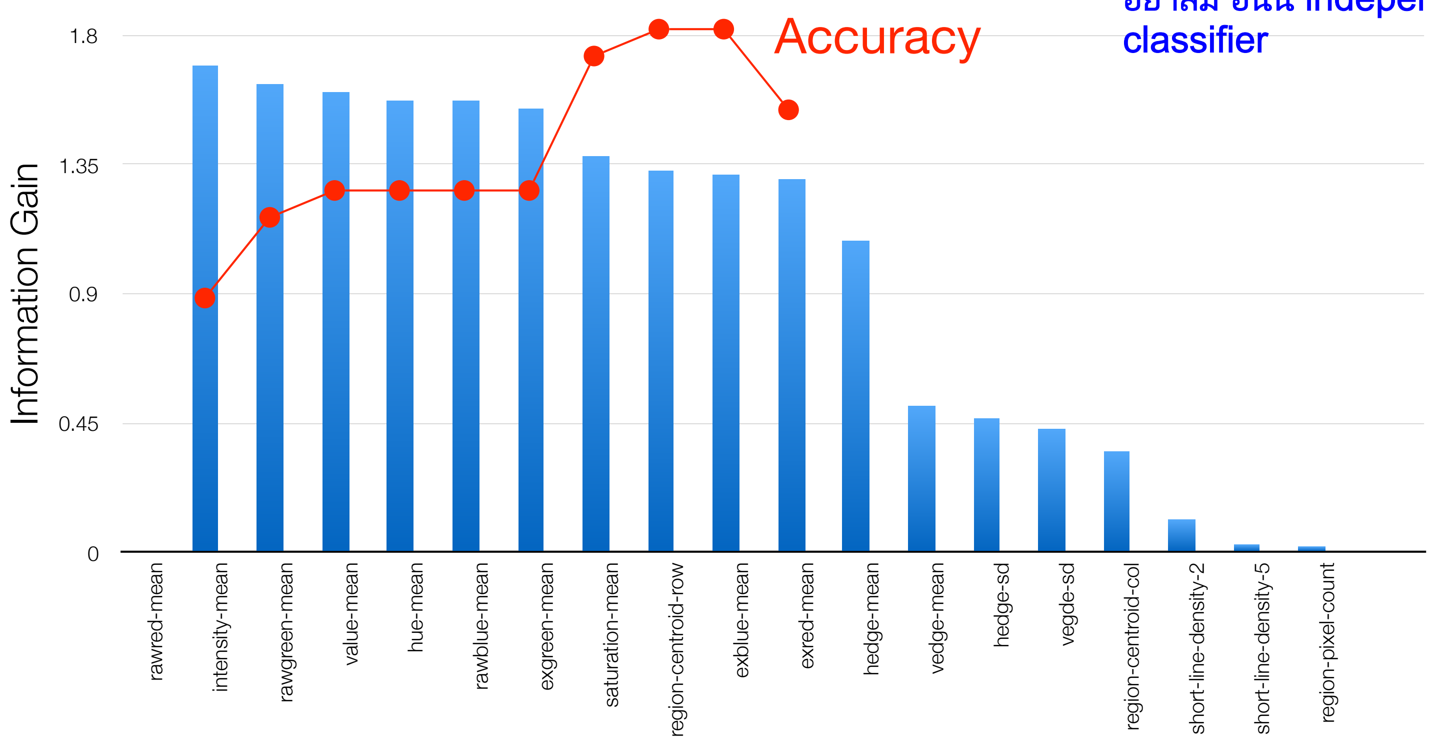
Filters - Top Features

- What to do with ranked list of features? Several options...
  - Select the top ranked k features.
  - Select top 50%.
  - Select features with IG > 50% of max IG score.
  - Subset of features with non-zero IG scores.



Filters - Top Features

- Common strategy for selecting k top features - test accuracy of subsets of increasing size.
- Start with feature with highest Information Gain, add next feature.
- Test accuracy for each subset using cross-validation.



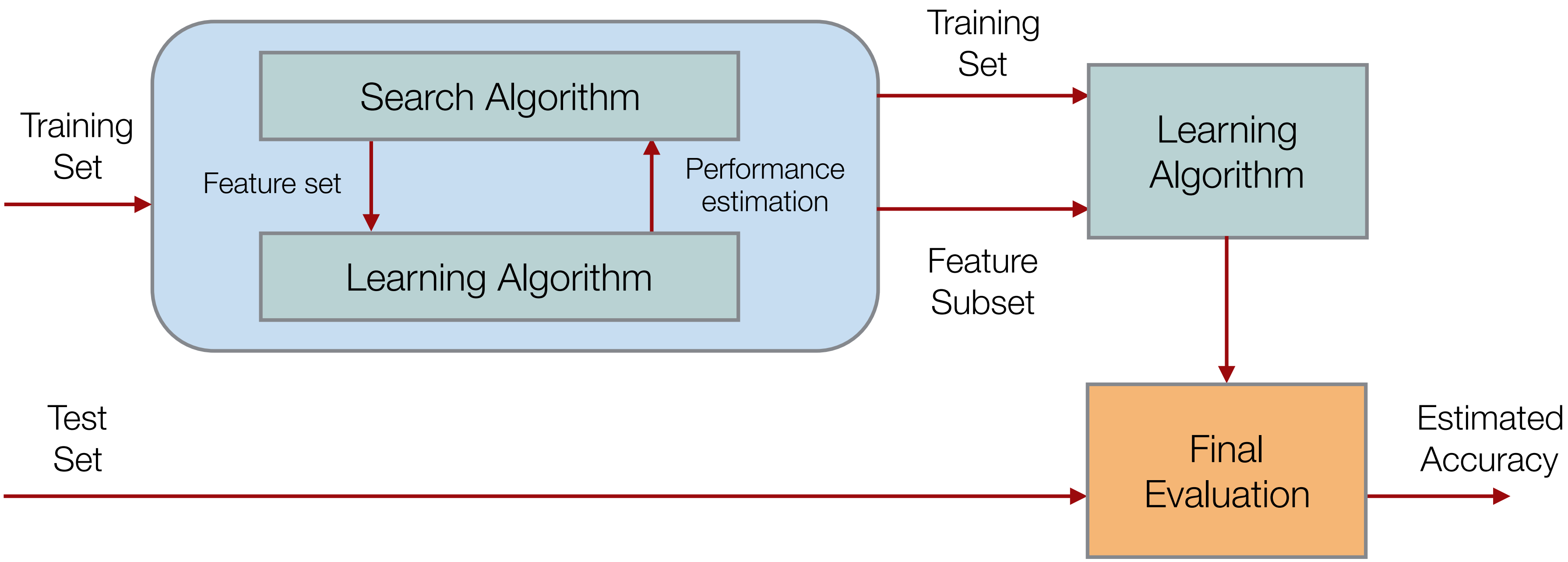
ขึ้นอยู่กับการมีเวลาในการทดลอง
อย่าลืม อันนี้ independent กับ
classifier

Filters - Disadvantages

- Problems with filter feature selection approaches:
  - **Model Bias** (model แต่ละตัวมันอิสระต่อกัน เวลาเราจะใช้เรา
ควรจะคิดถึง classifier และ model ที่จะใช้ให้ดี)
    - Different features may suit different learning algorithms (Neural networks, Decision Trees, K-NN, etc.).
  - **Dependencies**
    - Features are considered in isolation from one another, not considered in context.
    - In some cases, a filter might select two predictive but correlated features, where one would be sufficient.
    - In other cases, one feature needs another feature to boost accuracy.

Wrappers

- Alternative strategy: the classifier is “wrapped” in the feature selection mechanism. Feature subsets are evaluated directly based on their performance when used with that specific classifier.
- Advantages:
  - Takes bias of specific learning algorithm into account.
  - Considers features in context - i.e. dependencies.



search algorithm

Feature Search

- A key aspect of wrappers is the search strategy which generates the candidate feature subsets for evaluation.
- Different search strategies can generate different subsets. The choice of a suitable algorithm often depends on the number of features *d*.
  - Exhaustive search: Evaluate every possible subset of features. For *d* features there are 2^d potential subsets. For even small values of *d*, an exhaustive search over this huge space is intractable.
  - Exponential search: Returns the optimal feature subset, but uses heuristics so that not every one of the 2^d possible subsets needs to be evaluated. บางตัว (แบบย่อยจาก brute force) ทำให้อาจจะพลาดตัวดีไป
  - Sequential search: Fast search algorithms that choose a subset by adding or removing one feature at a time. Not guaranteed to final the optimal feature subset, but widely used.

Sequential Search

- Sequential search algorithms use a heuristic stepwise approach.
  - Forward Sequential Selection
    - Start with an empty subset.
    - Find the most informative feature and add it to the subset.
    - Repeat until there is no improvement by adding features.
  - Backward Elimination

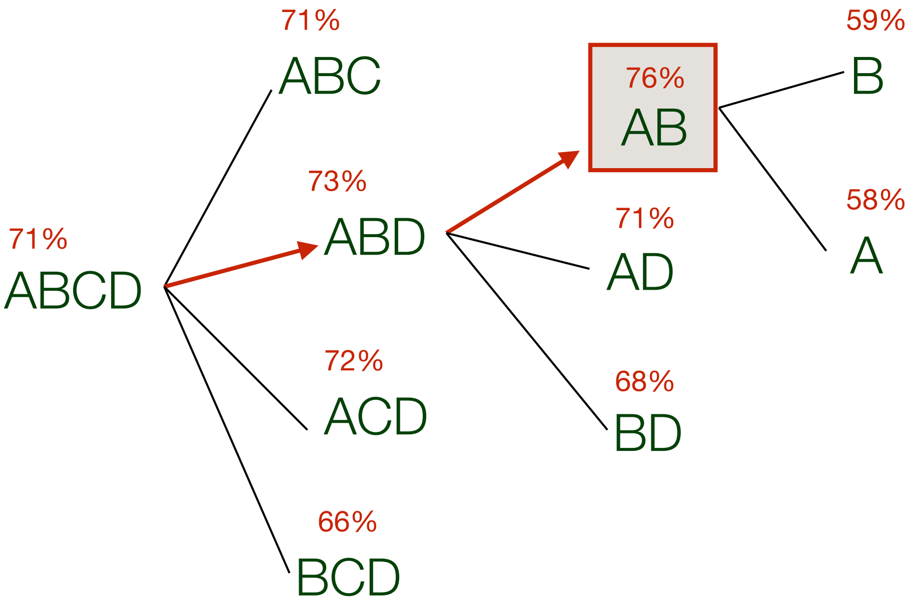
    เอาออกทีละ จน accuracy สูงเรื่อยๆ พอหลังจากสูงไปแล้วเริ่ม drop ก็จะตัดที่ตรงนั้น แล้วพอ

    - Start with the complete set of features.
    - Remove the least informative feature.
    - Repeat until there is no improvement by dropping features.

- Backward elimination tends to find better models - can find subsets with interacting features.
- Forward selection starts with small subsets, so less require less running time if stopped early.

Wrappers

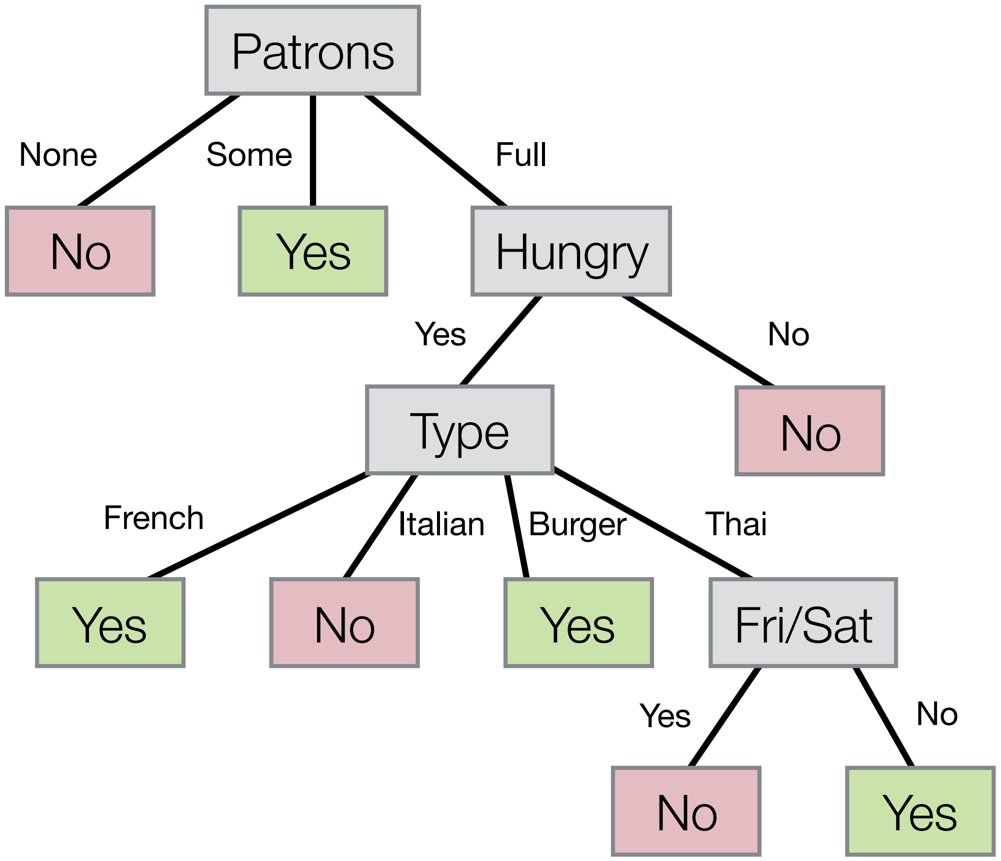
- Performance estimation (classification accuracy) guides the sequential search process.
- For robustness, performance estimation should use k-fold cross validation.
- Requires retraining the model main different times, can be very time consuming.
- Wrappers usually far slower than filters, but trying to solve the “real problem”.



Example: Backward elimination from 4 features (ABCD) to 2 features (AB).

Embedded Feature Selection

- Some methods do not even look like feature selection as they are "embedded" directly into the learning algorithm.
- Example: Decision Trees have a feature selection mechanism as an integral part of their core operation.



ID3 algorithm applies a feature selection + splitting process until all examples have the same class, or no features are left to split.

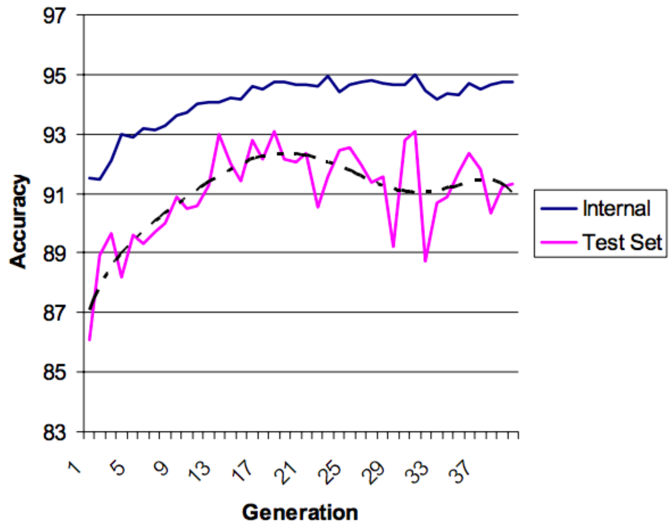
Overfitting in Feature Selection

- Reminder: A classifier is said to overfit to a data set if it models the training data too closely, leading to poor predictions when applied to unseen data.
- Wrappers can be prone to overfitting when the chosen feature subsets are used to build classifiers on unseen data.

➡ We simply found the best feature subset for the training data!

"Overfitting in feature selection appears to be exacerbated by the intensity of the search, since the more feature subsets that are visited the more likely the search is to find a subset that overfits"
- Loughrey & Cunningham, 2004

Example: Feature Selection with Genetic Algorithm Search



References

- R. Bellman. “Adaptive control processes: a guided tour”. Princeton University Press, 1961.
- E. Alpaydin. "Introduction to Machine Learning", Adaptive Computation and Machine Learning series, MIT press, 2009.
- P. Flach. “Machine Learning: The Art and Science of Algorithms that Make Sense of Data”. Cambridge University Press, 2012.
- K. Kira, L. Rendell. “A practical approach to feature selection”. Proc 9th *i*nternational workshop on Machine Learning, 1992.
- T. Mitchell. “Machine Learning”. McGraw-Hill, 1997.
- J. Loughrey, P. Cunningham. “Overfitting in Wrapper-Based Feature Subset Selection: The Harder You Try the Worse it Gets”. SGAI 2004