practical advice

selecting features

- dimensionality reduction

  algorithm? decision trees can do this, (select most promising feature at each node)

  problem but for smaller sets, even random attribute can look good with little data by chance

  feature selection是将数据投影到一个低维子空间上，是降维的一种方法；

  除了feature selection，dimensionality reduction方法还有kernal PCA（映射到高维空间）等；

- 最简单的：principal component analysis

  first principal component

  -> direction of the largest variance

  subsequent principal component

  -> orthogonal to the provious ones &

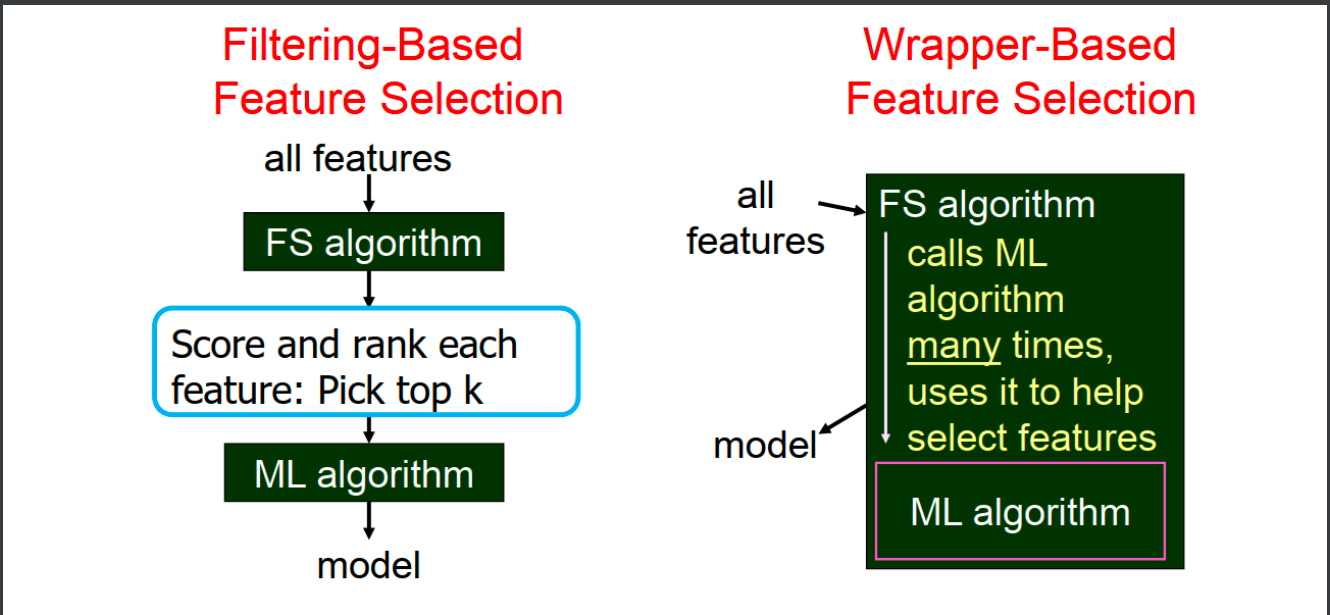
  directions of the largest variance of the residuals

  旋转坐标轴，使数据投影到方差最大的方向

  BUT,

  PCA cannot capture NON-LINEAR structure !!

- 2 approaches: filter-based / wrapper-based



可以理解为filtering-based只用统计方法或指标来评估特征，在selection过程中不涉及具体model的学习

wrapper-based是指搜索特征子集，需要一个具体的机器学习模型；

filter-based approaches

measure each feature's usefulness in isolation

pro	cons	
very fast, scales to larger feature sets / data sets	misses feature interactions	
	might select redundant features	

1. correlation

   for example, information gain, or standard correlation:

$$R(f_i, y) = \frac{\sum_{k=1}^{m}\left(f_{k,i} - \overline{f}_i\right)\left(y_k - \overline{y}\right)}{\sqrt{\sum_{k=1}^{m}\left(f_{k,i} - \overline{fi}\right)^2 \sum_{k=1}^{m}\left(y_k - \overline{y}\right)^2}}$$

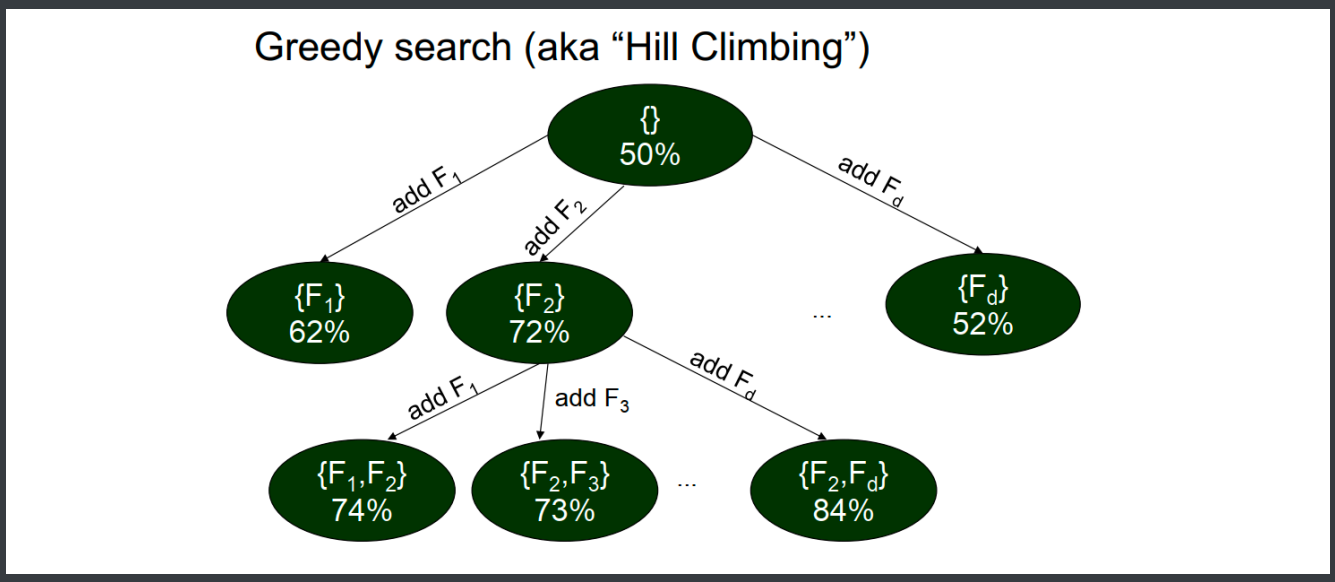
2. single variable classfier

   build classifier with just one variable

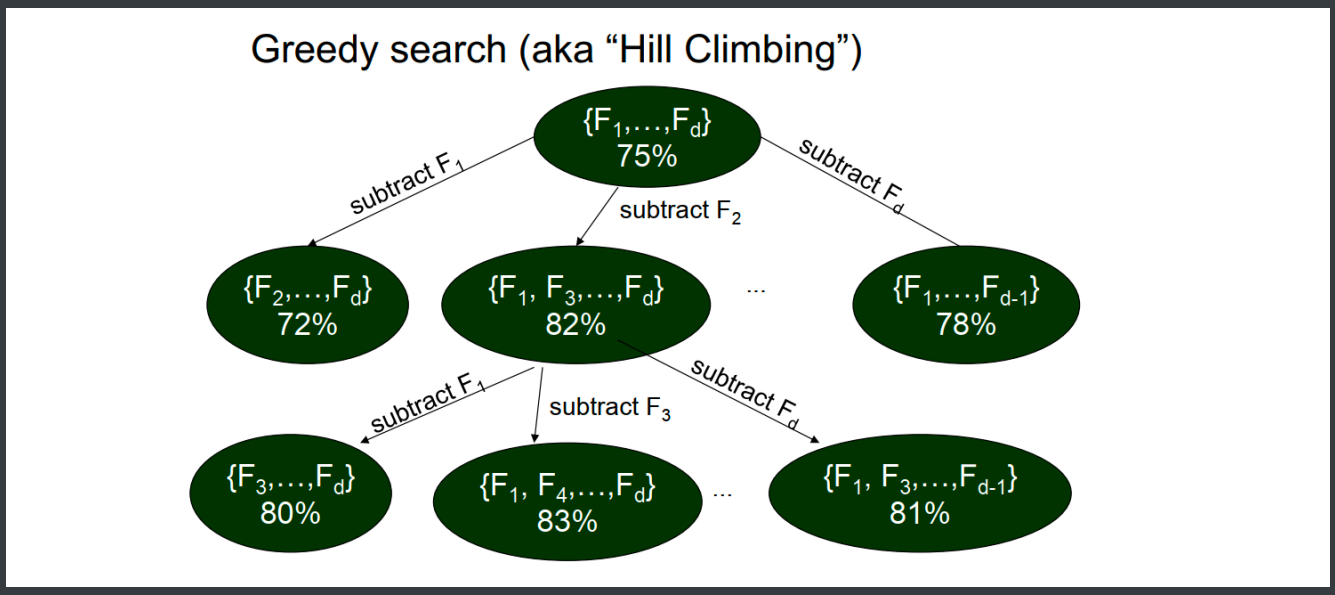
wrapper-based approaches

feature selection = search

- forward selction: start from empty



- backward selection: start from full



- backward preserves features whose usefulness requires other features

advice for evaluation

- comparing run times is a dark art
- ablative analysis 消融实验

  remove aspects of systems and measure effect on performance

potential problems or pitfalls

cross validation errors

- only training data!!
- use small scale of data for development, then expand evaluation
- **temporal dependencies** in the data?

class imbalance

problem: often more examples of one class than the other

e.g. cancer detection, always more negative detection

solution 1: sampling

generate synthetic minority examples (interpolate between nearest neighbors)

solution 2: manipulate the learner

change the cost function, penalize mistakes on minority class more heavily

optimize towards something that is better at capturing skew

model is not accurate enough

possible fixes:

more data, more / better features (important!!!)

optimizer, objective function, model class