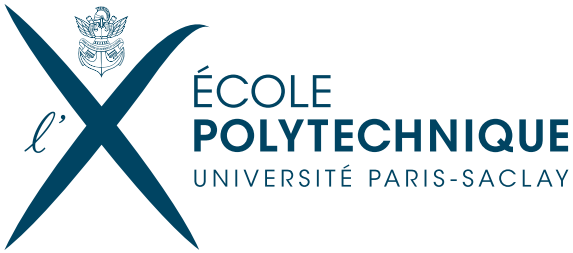
Machine Learning - methods, feature selection, evaluation and VC-dimension

M. Vazirgiannis,
Google Scholar Profile: https://bit.ly/2rwmvQU



October, 2024

Outline

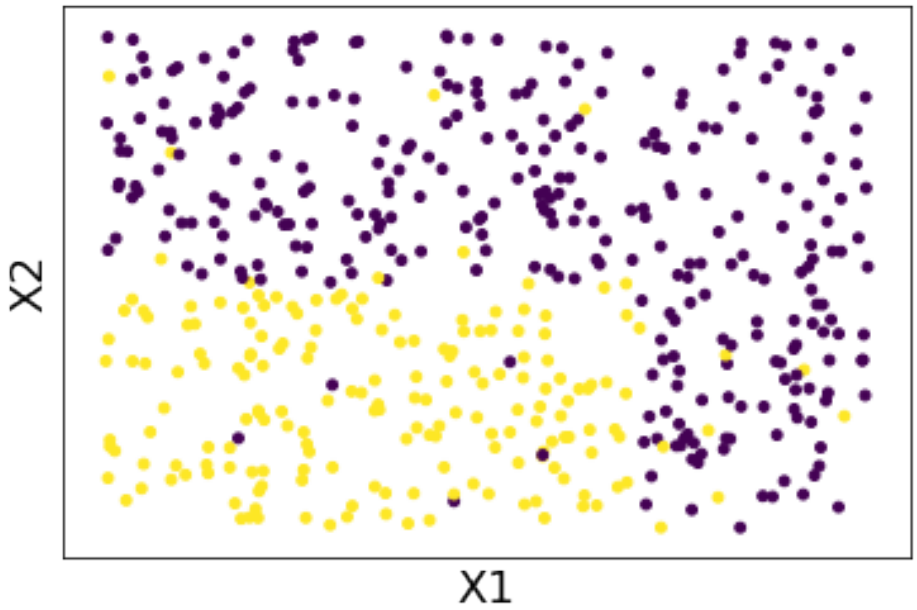
Decision Tree Classifiers

- Widely used in practice
  - can handle both real-valued and nominal inputs
  - good with high-dimensional data
- Historically, developed both in statistics and computer science
  - Statistics
    - Breiman, Friedman, Olshen and Stone, CART, 1984
  - Computer Science
    - Quinlan, ID3, C4.5 (1980's-1990's)

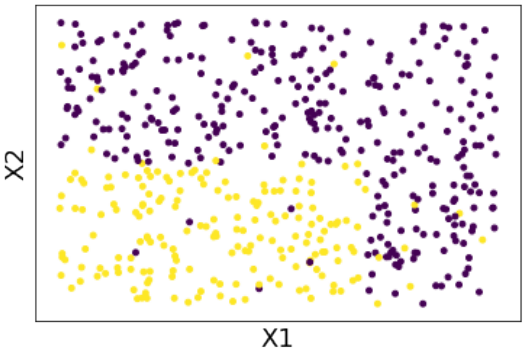
Intuition on Data Splitting

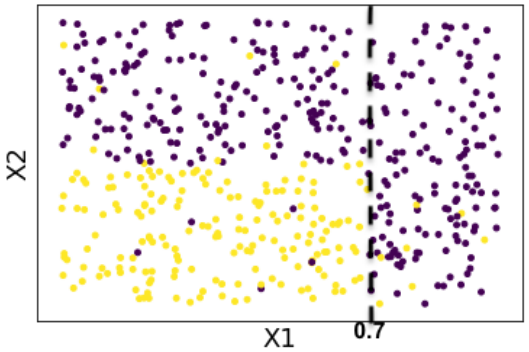
- Data X, set of attributes F where x_{ij} value of feature j of data point i.
- Assume $Y = y_k$ class labels.
- Objective: find a partition of the data based on the distinct feature values resulting in clusters of homogeneous class labels (purity of classes)

Devide and Conquer

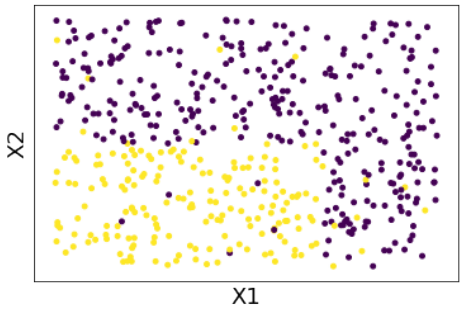


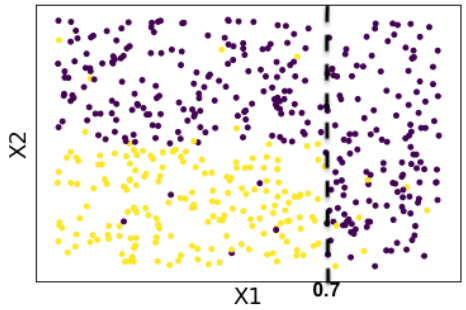
Devide and Conquer

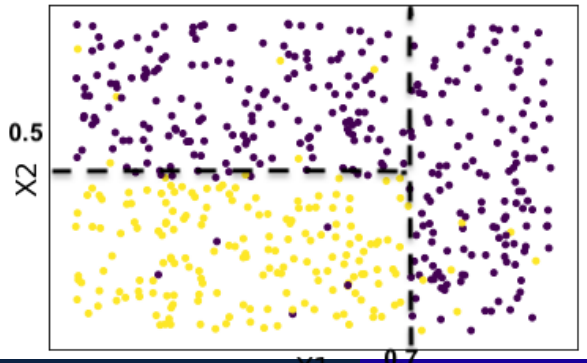




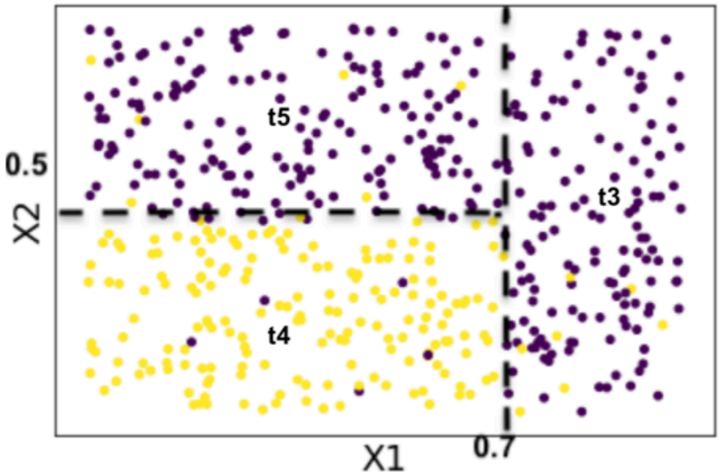
Devide and Conquer

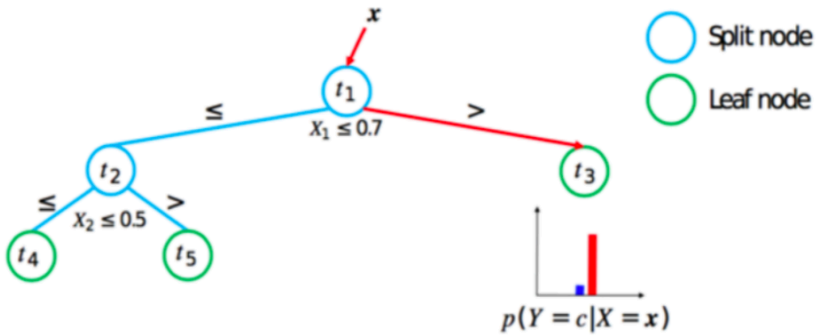






Decision Tree





$t \in \phi$: nodes of the tree ϕ

X_t : split variable at t

$v_t \in \mathbb{R}$: split threshold at t

$\phi(x) = argmax_{c \in y} p(Y = c|X = x)$: split threshold at t

Node Impurity

- Need a splitting criterion
- Assume an impurity measure F (i.e. # of miss-classified data points)
- In each next round, a node n_t is split based on some question q_t. The pair (n_t, q_t) is chosen so that the node impurity is maximally decreasing according to some measure of impurity F.

Learning

- Learning the simplest (smallest) decision tree:
  NP-complete [Hyafil & Rivest '76]
- Use a greedy heuristic:
  - all data in a node (root)
  - split on **next best attribute** (feature)
  - recurse
- Algorithm stops in either case
  - all nodes reached a sufficient level of purity
  - # of nodes/leaf became too small for further splitting
  - some other similar heuristic.

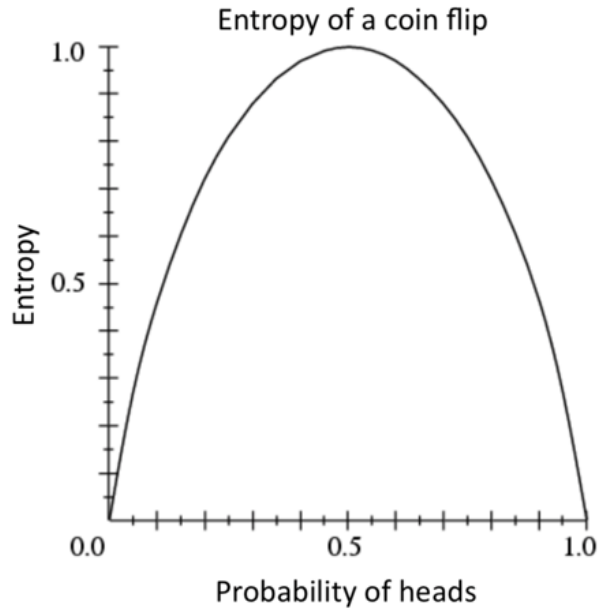
Node Metrics

- The three most commonly used measures of node impurity F

$$F(n) = \begin{cases} 1 - max_{l \in [1,k]} p_l(n), & \textbf{misclassification} \\ -\sum_{l=1}^{k} p_l(n) log_2 p_l(n), & \textbf{entropy} \\ \sum_{l=1}^{k} p_l(n)(1 - p_l(n)), & \textbf{Gini index} \end{cases} \quad (1)$$

Entropy-Information Gain

- Assume a variable X
- Its value set $x_1, ...x_k$
- The entropy of X
  $H(X) = \sum_1^k p(x_i) log_2(p(x_i))$
- Assume conditioning X on different features



$$H(X|A_i) = \sum_{j=1}^{k} p(x_j|A_i) log_2(p(x_j|A_i))$$

- Information gain for A_i

$$IG(A_i) = H(X) - H(X|A_i)$$

Information Gain (ID3/C4.5)

- Select the attribute with the highest information gain
- Assume there is a class P
  - let the set of examples S contain p positive elements of class P and n negative elements.
  - the entropy of the class distribution is:

$$I(p,n) = -\frac{p}{p+n} log_2 \frac{p}{p+n} - \frac{n}{p+n} log_2 \frac{n}{p+n}$$

Example

$p(Cancer) = 0.25$

$p(Sports) = 0.416, p(Smoke) = 0.33$

$p(C|Sports) = 0.2, p(C|Smoke) = 1$

SPORTS	SMOKE	CANCER
0	1	1
0	0	0
1	1	0
0	0	0
0	0	0
1	1	1
1	0	0
0	1	1
0	0	0
1	0	0
1	0	0
0	0	0

Example

SPORTS	SMOKE	CANCER
0	1	1
0	0	0
1	1	0
0	0	0
0	0	0
1	1	1
1	0	0
0	1	1
0	0	0
1	0	0
1	0	0
0	0	0

$p(Cancer) = 0.25$

$p(Sports) = 0.416, p(Smoke) = 0.33$

$p(C|Sports) = 0.2, p(C|Smoke) = 1$

$H(Cancer) = -p(C)log(p(C)) - (1 - p(C)log(1 - p(C))) = 0.81$

$H(Cancer|Sports) = -p(C|S)log(p(C|S)) - (1 - p(C|S)log(1 - p(C|S))) = 0.722$

$H(Cancer|Smoke) = -p(C|Sm)log(p(C|Sm)) - (1 - p(C|Sm)log(1 - p(C|Sm))) = 0.0001$

Example

SPORTS	SMOKE	CANCER
0	1	1
0	0	0
1	1	0
0	0	0
0	0	0
1	1	1
1	0	0
0	1	1
0	0	0
1	0	0
1	0	0
0	0	0

$p(Cancer) = 0.25$

$p(Sports) = 0.416, p(Smoke) = 0.33$

$p(C|Sports) = 0.2, p(C|Smoke) = 1$

$H(Cancer) = -p(C)log(p(C)) - (1 - p(C)log(1 - p(C))) = 0.81$

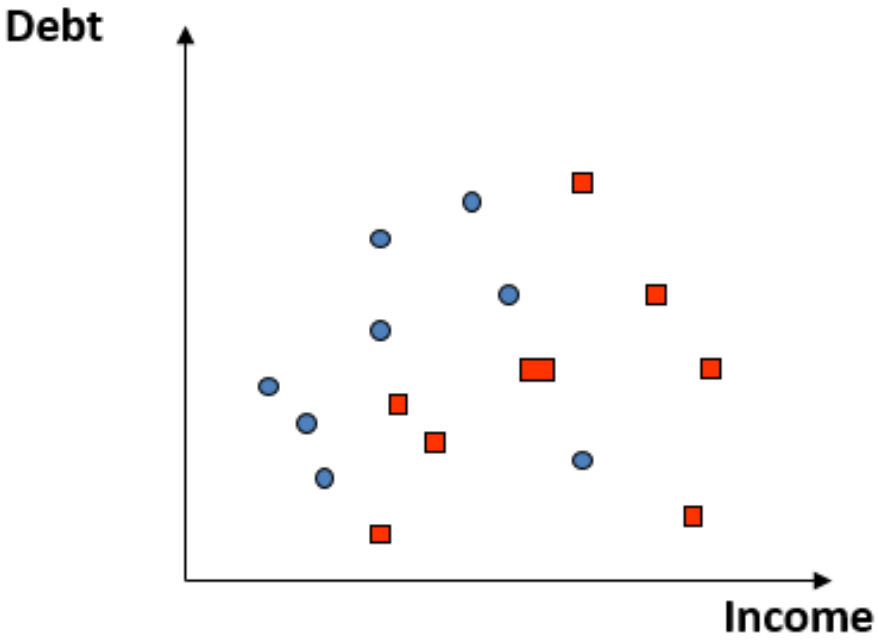
$H(Cancer|Sports) = -p(C|S)log(p(C|S)) - (1 - p(C|S)log(1 - p(C|S))) = 0.722$

$H(Cancer|Smoke) = -p(C|Sm)log(p(C|Sm)) - (1 - p(C|Sm)log(1 - p(C|Sm))) = 0.0001$

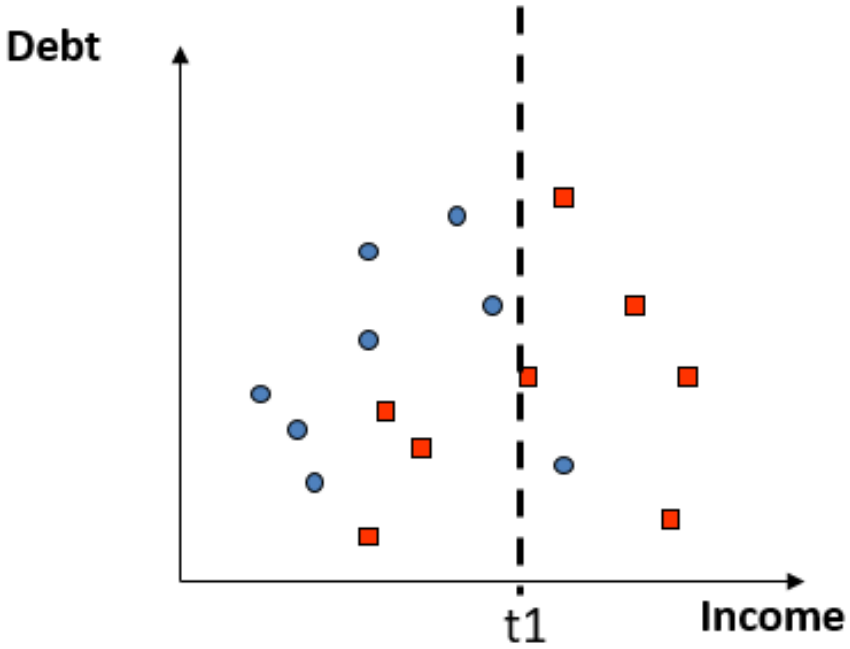
$IG(S_m) = H(Cancer) - H(Cancer|Sm) = 0.809$

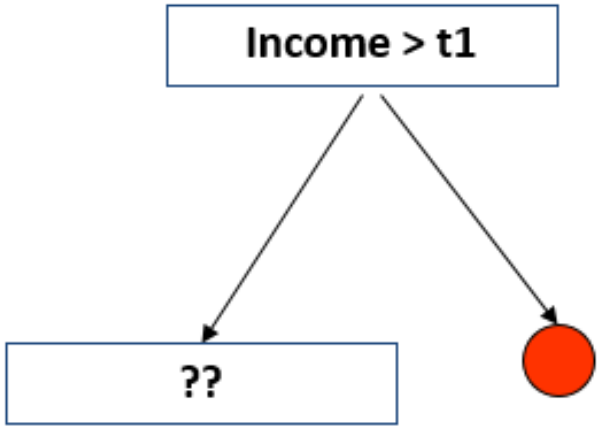
$IG(S) = H(Cancer) - H(Cancer|S) = 0.098$

Decision Tree Example

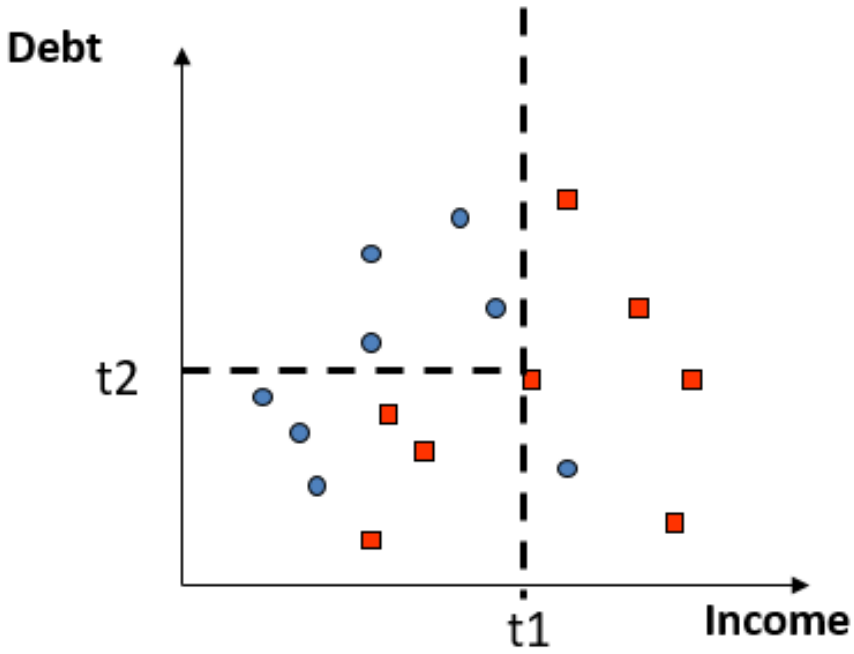


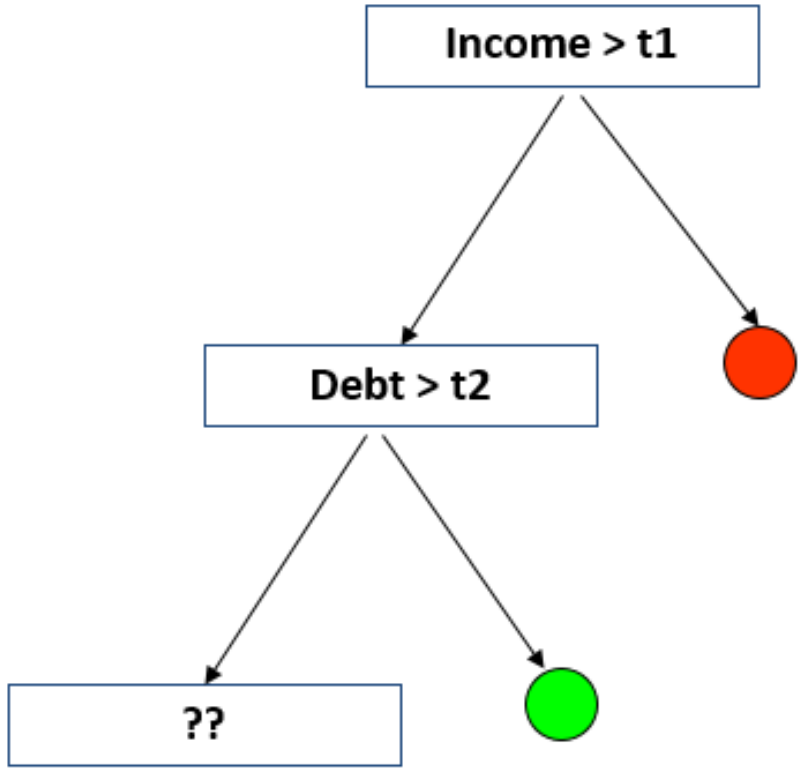
Decision Tree Example



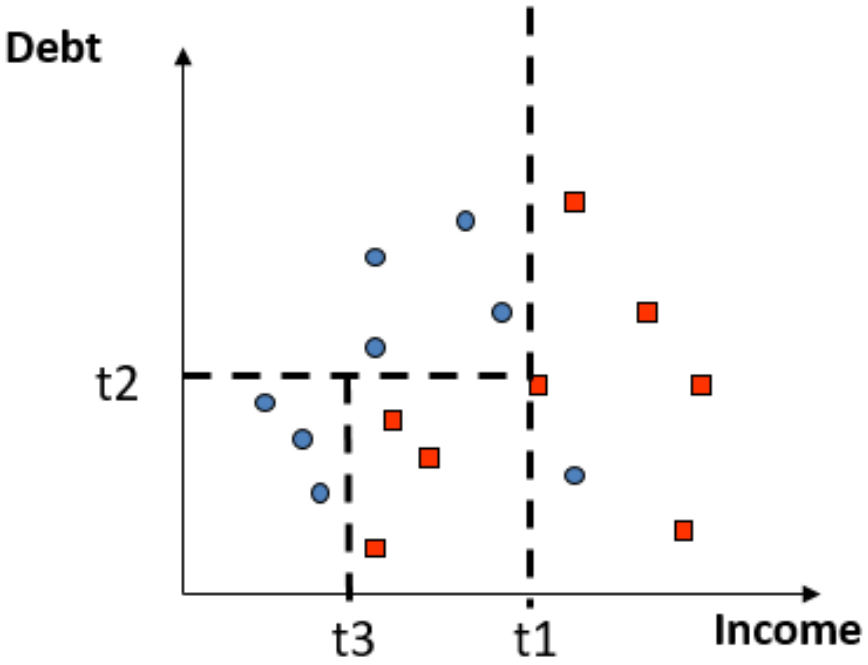


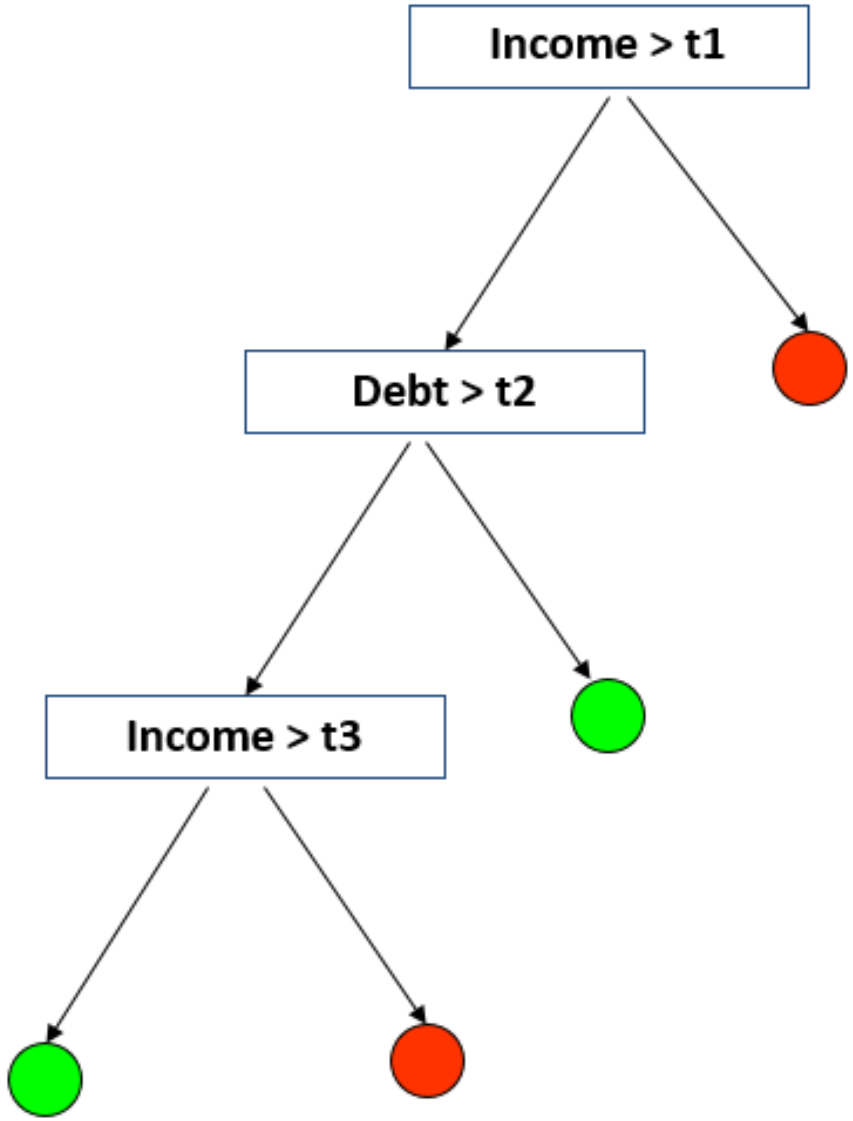
Decision Tree Example



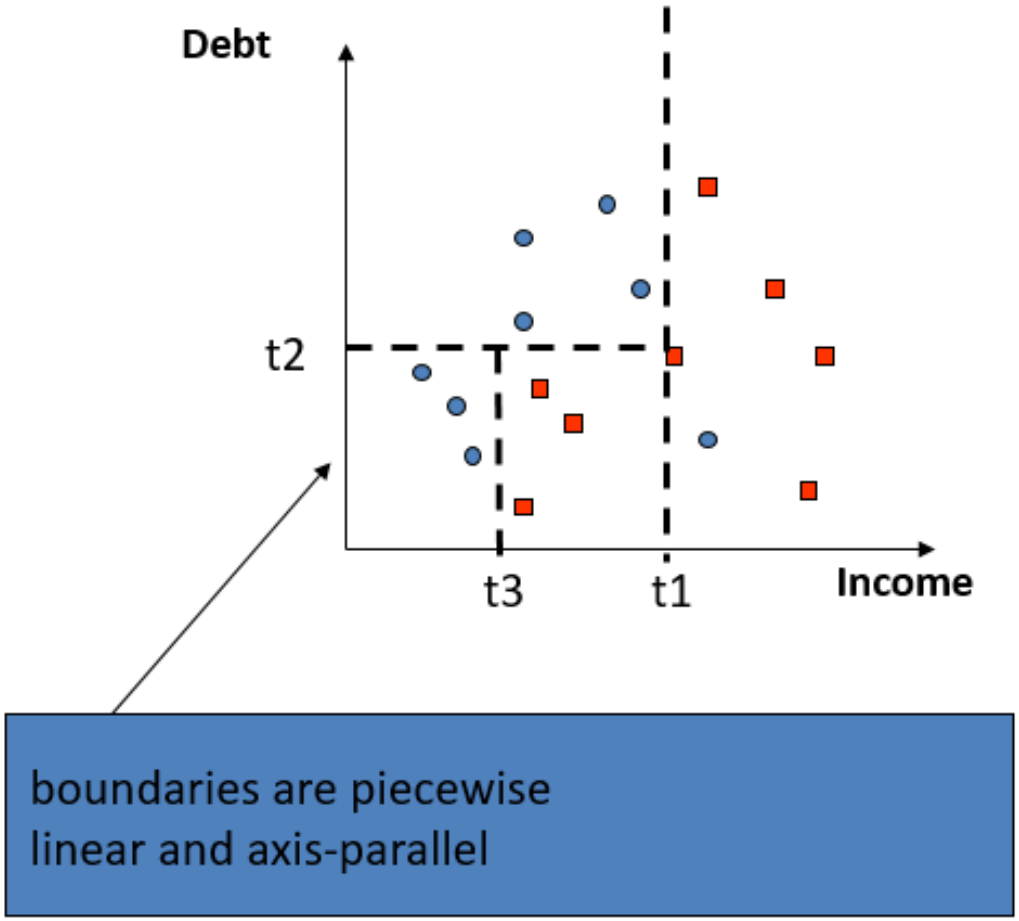


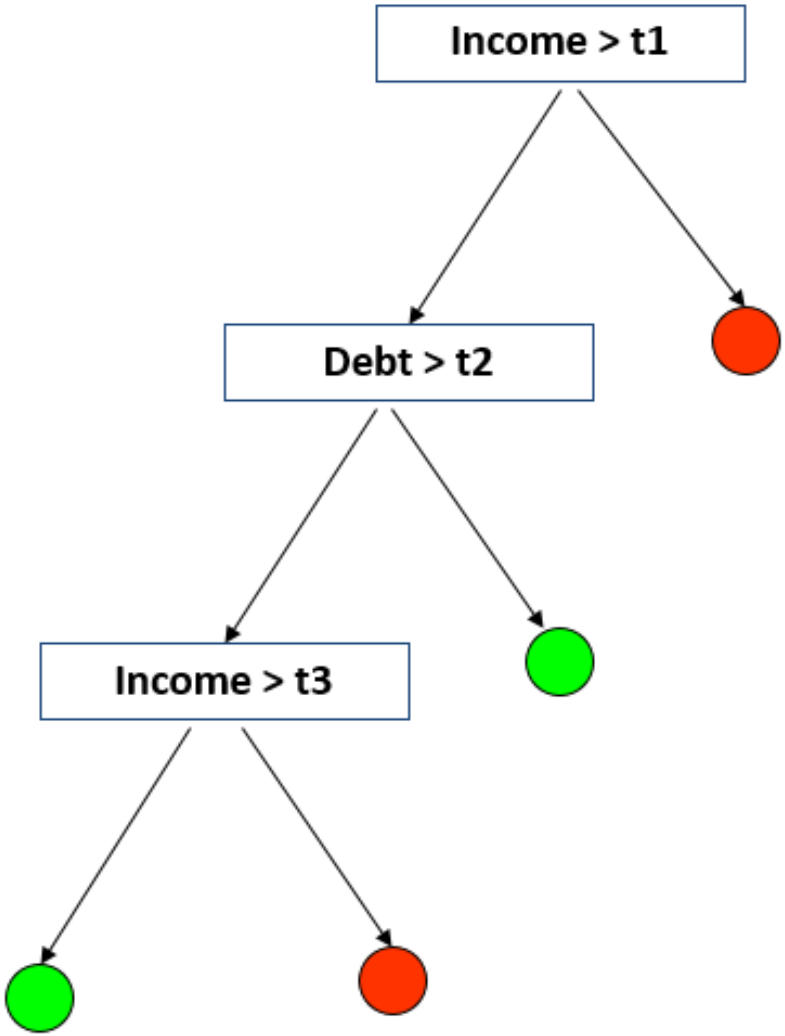
Decision Tree Example





Decision Tree Example

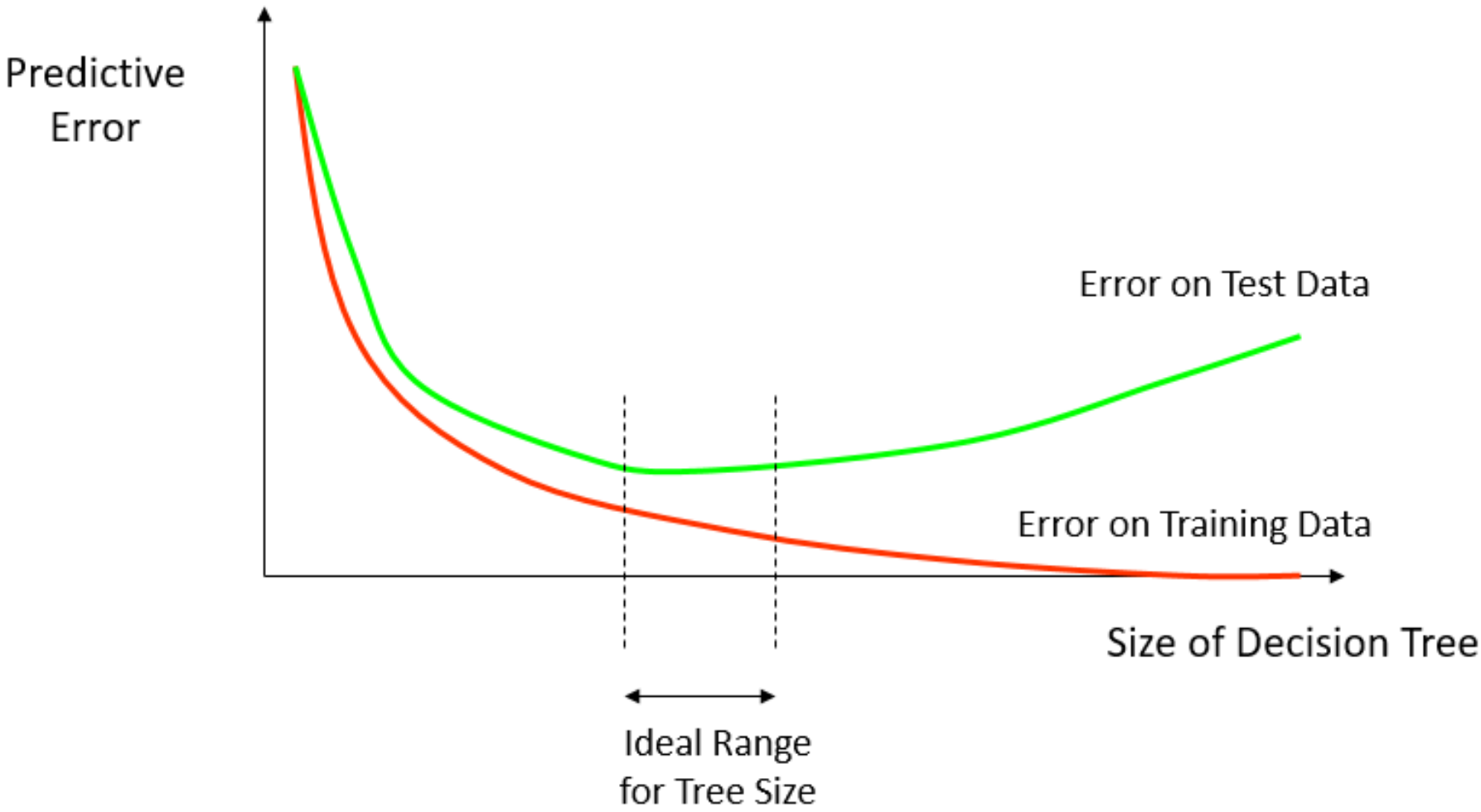




DT: Considerations

- When do we stop training?
  - everything is classified correctly
  - no more attributes to train on
  - no overfitting (ocams razor)
    - CV?

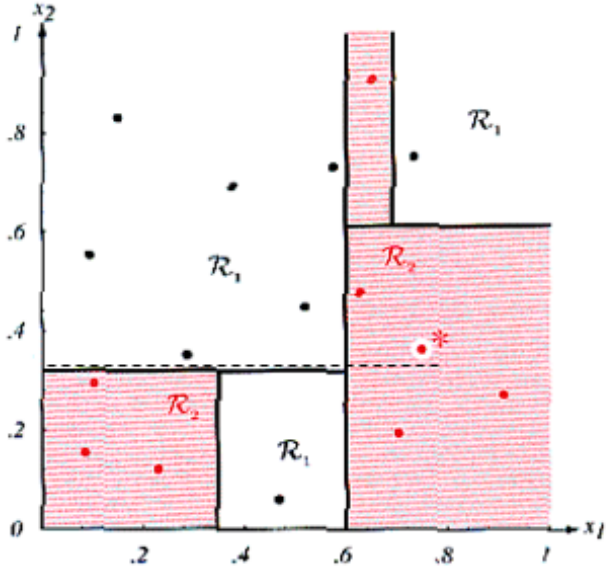
How to Choose the Right-Sized Tree?

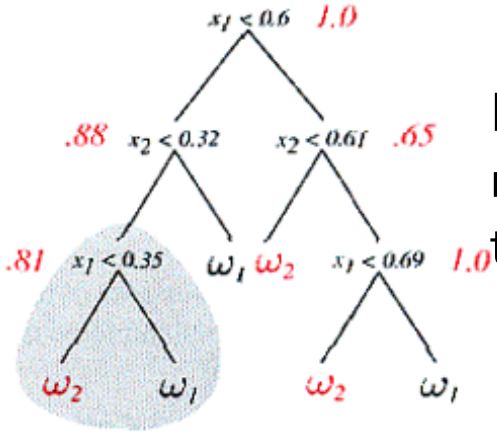


Why Trees are widely used in Practice

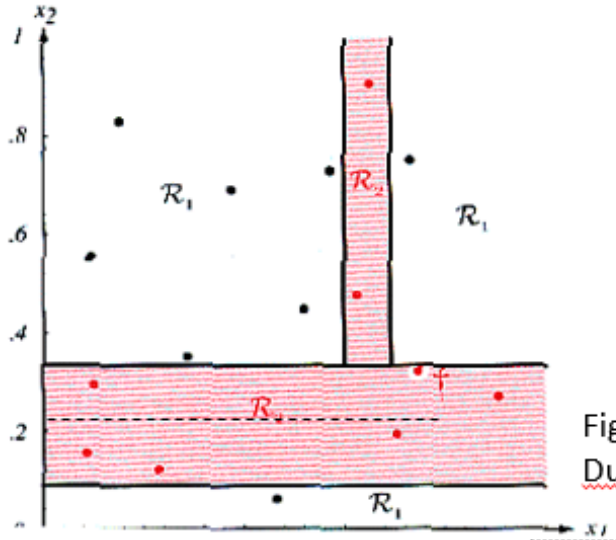
- Can handle high dimensional data
  - builds a model using 1 dimension at time
- Can handle any type of input variables
  - categorical, real-valued, etc
  - most other methods require data of a single type (e.g., only real-valued)
- Trees are (somewhat) interpretable
  - domain expert can "read" the tree's logic
- Tree algorithms are relatively easy to code and test

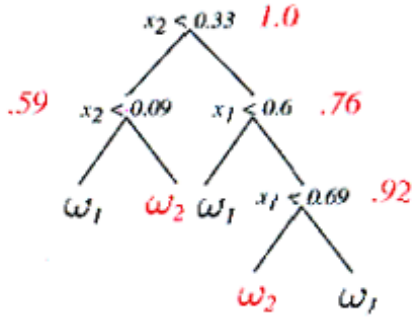
Decision Trees are not stable





Moving just one example slightly may lead to quite different trees and space partition!





Lack of stability against small perturbation of data.

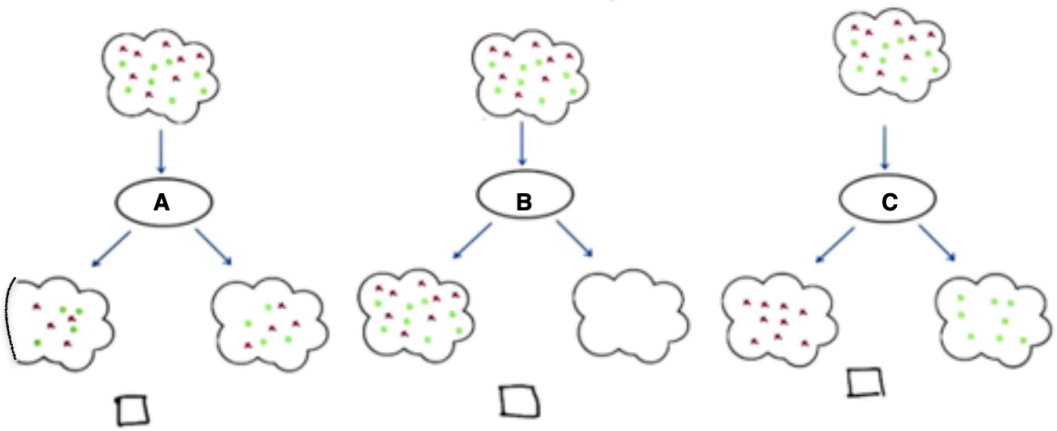
Figure from
Duda, Hart & Stork, Chap. 8

Limitations of Trees

- Greedy optimization
- Hard decision boundaries
  - classification: piecewise linear boundaries, parallel to axes
  - regression: piecewise constant surfaces
- Over fit their training sets; low bias, high variance.
- High Variance
  - trees can be "unstable" as a function of the sample
    - e.g., small change in the data $\rightarrow$ completely different tree
  - causes two problems
    - contributes to prediction error
    - reduces interpretability
- Solution : Combine the predictions of several randomized trees into a single model.

Quiz

1. Which is the best attribute for splitting?



2. Does it make sense to have same variable multiple times in a decision tree?

Random forests

- Ensemble learning method for classification, regression
- Construct a set of decision trees at training time
- Output the class with majority of the classes (classification) or mean prediction (regression) of the individual trees.
- Random decision forests heal the overfitting of decision trees to the expense of increased bias

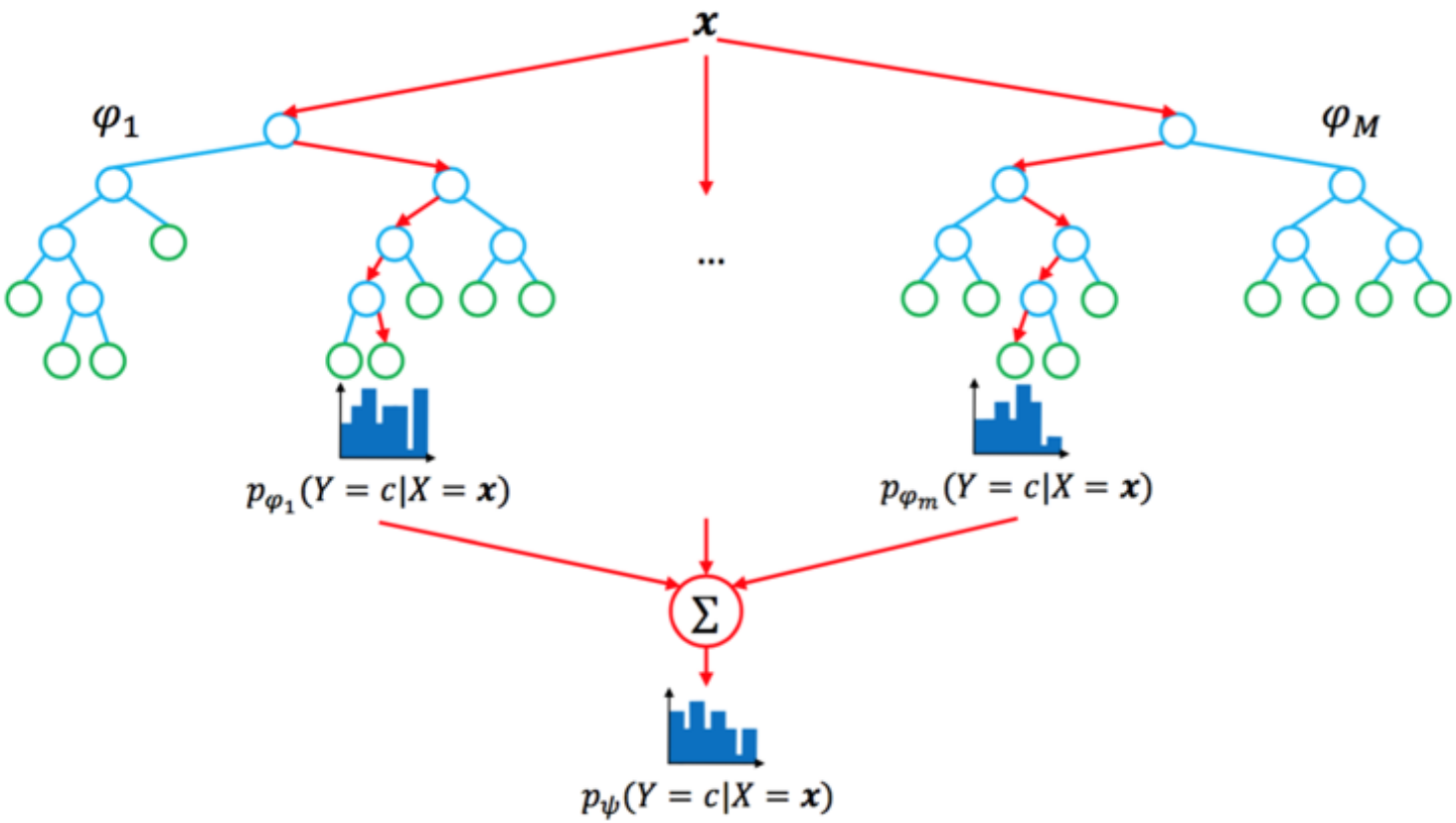
Bootstrap Aggregating - Bagging

- Assume training set $D = \{(xi, yi)\}$
- Objective: predict label for an unknown x
  - sample B data sets – each of size n, randomly with replacement from D: $D_1, \ldots D_B$
  - for each D_i train a tree and make a prediction to obtain a set of B predictions
  - final prediction obtained by averaging (regression) or majority voting (classification)
- Decreases variance in the predictions
- B is a free parameter: An optimal number of trees B can be found using cross-validation

Random Forests

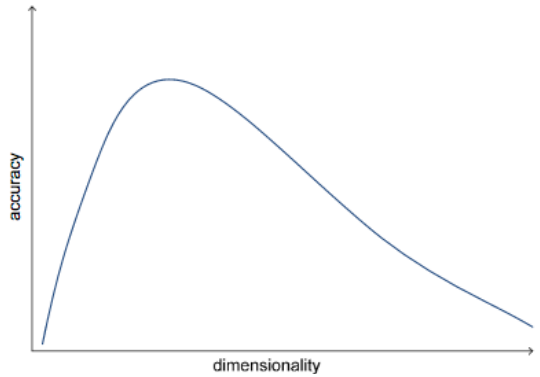
- Random forests use a modified tree learning algorithm selects, at each candidate split in the learning process, a *random subset of the features* - *"feature bagging"*.
- Reason - correlation of the trees:
  - if some features are very strong (i.e. high Information Gain) they will be selected in many of the B trees,
  - resulting trees correlated.
- # of features selected:
  - classification problem with p features, $\sqrt{p}$ features are used in each split .
  - regression problems have different defaults

Random Forests



Outline

Dimensionality vs. Accuracy



- reduced dimensionality corresponds to the intrinsic dimensionality of the data.
- minimum number of independent variables needed to explain the observed properties of data.

Dimensionality reduction

- select the most discriminative features
- low-dimensional data representations imply a physical meaning.

Feature extraction.

- create new informative features by transforming the original features.
- new projection of data based on transformation or combination of the original feature set. i.e. SVD, PCA, MDS, NMF

Feature Selection

- "principle of parsimony" [Bell and Wang, 2000] : prefer model with the smallest possible number of parameters that adequately represents the data.
- Feature selection methods aim at selecting an optimal subset of relevant features from a given set of original candidate features

Issues

- feature subset generation (or search strategy);
- evaluation criterion definition (e.g. relevance index or predictive power);
- evaluation criterion estimation (or assessment method).

Feature Selection

Select the best features (subset of the original one)

- Univariate, Filter methods: rank individually features based on some criterio(i.e. inf. gain,χ^2, etc) and select the top-k features
- Wrapper methods: evaluate each subset of features. use heuristics for the exploration (forward / backward search)
- Embedded methods: feature selection is part of the ML algorithm

Feature Selection

Univariate methods: rank features according to their individual relevance

- fast and effective - number of features large, number of training examples small (e.g. 10,000 features and 100 examples.)

Problems due to feature in/dependence

- features not individually relevant may become relevant in the context of others
- features individually relevant may not all be useful because of possible redundancies.

Univariate methods - Information Gain (IG)

For a random variable X (class) its entropy :

$$-\sum_{i=1}^{c} p(x_i) log(p(x_i)), c\text{: number of classes}$$

- "High Entropy": x is from a uniform distribution – lack on information
- "Low Entropy": x is from varied (peaks and valleys) distribution – rich in information content

Let variable A (feature), $IG(x, A)$ represents reduction in entropy (gain in Information) of X achieved by learning the state of A:

$$IG(x, A) = H(x) - H(x|A)$$

- features not individually relevant may become relevant in the context of others
- features individually relevant may not all be useful because of possible redundancies.

Univariate methods -Pearson correlation

- Relevance index
- letx_j $m-$dimensional vector containing all the values of the j-th feature for all training examples
- let y $m-$dimensional vector containing the target values. Then:

$$c(j) = \frac{|\sum_{i=1}^{m}(x_{i,j}-\bar{x})(y_i-\bar{y})|}{\sqrt{\sum_{i=1}^{m}(x_{i,j}-\bar{x})^2(y_i-\bar{y})^2|}}$$

  - cosine similarity between feature x_j and decision y, after they have been centered.

Univariate methods -Chi-squared test (χ^2)

- Test of independence between a class X and a feature A

$$\chi^2(A) = \sum_{i=1}^{u} \sum_{=1}^{c} \frac{(o_{ij} - e_{ij})^2}{e_{i,j}}$$

u: A's values, c: classes

  - let o_{ij}: observed frequency of class j in for value i of feature A
  - let e_{ij}: expected frequency of class j in for value i of feature A

$$e_{ij} = \frac{(\#\text{samples with value i for A when class} = j)(\#\text{samples with value j for c}}{\text{total } \# \text{ samples}} \quad (2)$$

Univariate methods -The relief method

- Ranking index $C(j)$ and a feature j
- let x_i a data point and j-th feature under concern
- let $x_{H(i),j}$ - the k closest points to x_i of the class to which x_i belongs
- let $x_{M(i),j}$ - the k closest points to x_i of different classes (misses)

$$c(j) = \frac{\sum_{i=1}^{m} \sum_{f=1}^{k} |x_{i,j} - x_{M_{f(i),j}}|}{\sum_{i=1}^{m} \sum_{f=1}^{k} |x_{i,j} - x_{H_{f(i),j}}|}$$

Evaluation of filter methods

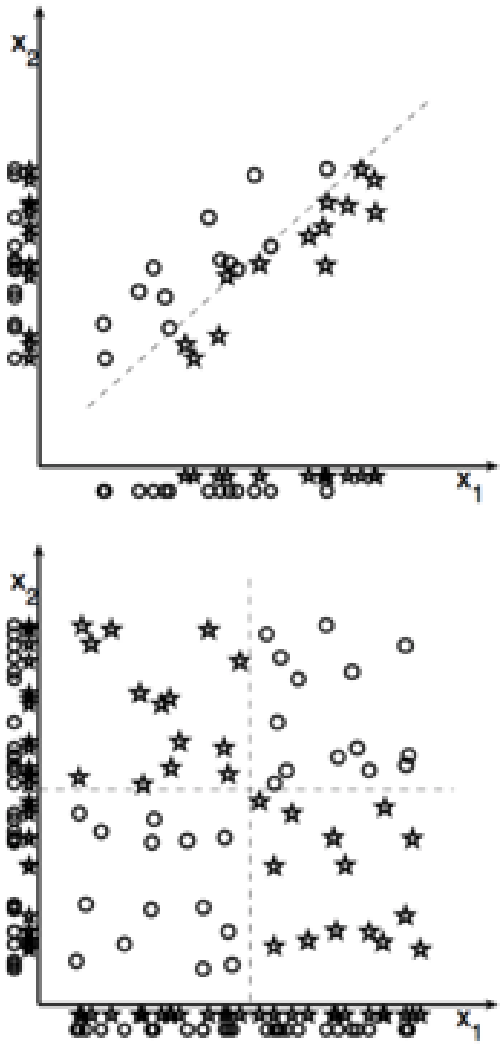
Advantages of filter techniques

- scale to high dimensional datasets.
- computationally simple and efficient
- independent of the algorithm.
- feature selection performed once, various classifiers can be evaluated.

Disadvantages

- Ignore interaction with the mining algorithm
- Search in feature subset space separated from search in hypothesis space - may lead to worse performance
- each feature considered separately, lack robustness against interactions among features and feature redundancy.

Multivariate methods - Correlation Impact on Variable Redundancy

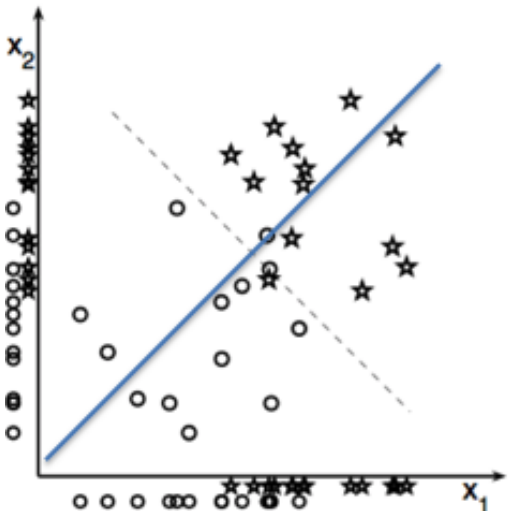


Isabelle Guyon, Andre Elisseeff, An Introduction to Variable and Feature Selection, Journal of Machine Learning Research, 3 (2003) 1157-1182

Relevant Features Individually Irrelevant

- Features individually irrelevant may become relevant when used in combination.
- linear separation where individually irrelevant features (x2) combined with an informative (x1) enable a better separation
- Two individually irrelevant features x2, x1 may become relevant when used together in combination.

Multivariate methods - Correlation Impact on Variable Redundancy

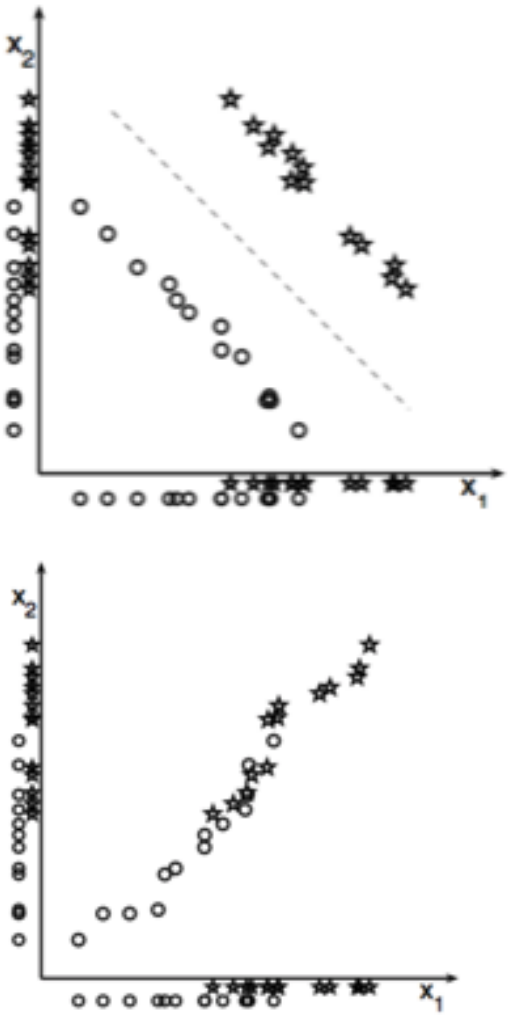


Isabelle Guyon, Andre Elisseeff, An Introduction to Variable and Feature Selection, Journal of Machine Learning Research, 3 (2003) 1157-1182

Relevant Features - Noise reduction

- features x_1, x_2 good for class separation
- their combination gives even better results.
- Assume classes generated by Gaussian distributions with equal variance σ^2.
- Project the data to each feature: distance d between classes centers is identical.
- signal to noise ratio of each individual feature d/σ.
- consider both features and project to the diagonal - distance between classes: $d\sqrt{2}$
- Adding n features with such class conditional independence results in an improvement of $\sqrt{n}$

Multivariate methods - Correlation Impact on Variable Redundancy



Isabelle Guyon, Andre Elisseeff, An Introduction to Variable and Feature Selection, Journal of Machine Learning Research, 3 (2003) 1157-1182

Relevant Features - Correlation

- features x_1, x_2 highly correlated but considering them both improves separability
- features anti-correlated – still both features give a better separation

Multivariate methods - Subset selection

- Assume n variables, find the subset of k variables for best prediction performance
- 2-variable models: $\frac{n(n-1)}{2}$ to evaluate
- ...
- in total we need to evaluate 2^n candidate models
- Best set of k variables $\neq$ set of best k individual variables: What does "best" mean here?

Multivariate methods - Subset selection - Search problem

- Assume n variables, need to evaluate 2^n candidate models: not feasible
- Sequential search is used to search over model space:
  - Forward search (greedy – hill climbing)
  - Backward search (greedy - hill climbing)
  - Branch and bound techniques
  - Variable selection problem in several data mining algorithms
    - Outer loop that searches over variable combinations
    - Inner loop that evaluates each combination

Multivariate methods - Subset selection -Sequential Search Strategy

- Wrappers: two most commonly used wrapper methods:
  - Sequential Forward Selection (SFS)
  - Sequential Backward Elimination (SBE)
- exploit a greedy hill-climbing search strategy.
- SBE
  - starts with all features
  - progressively eliminates the least promising ones
  - stops if the evaluated performance drops below a given threshold,
- SFS: adds features until performance stops improving.

Multivariate methods - Subset selection -Forward Selection

- Need a score to evaluate the feature set – i.e. the $p - value$ (i.e. $pvalue < 0,05$: evidence for rejecting the null hypothesis)
- Start with the variable with the lowest $p - value$
- assume two models m_2, m_1 with $|m_2| \geq |m_1|$, the full model contains M features.
- add in each repetition the variable with the highest $F - test$ value:

$$(\frac{RSS_{m_1} - RSS_{m_2}}{|m_2| - |m_1|})/(\frac{RSS_{m_2}}{M - |m_2|}) \tag{3}$$

- where $RSS = \sum_{i=1}^{n}(y_i - f(x))^2$

Multivariate methods - Subset selection -Forward Selection

- Backward Elimination
  - start with full model
  - Drop feature that produces the smallest F value (or highest p-value)
  - Continue until F-value $< F_{threshold}$ (or p-value$> p_{threshold}$)
- Bidirectional selection
  - search can start from both ends and iteratively add and remove features simultaneously [Huan and Hiroshi, 1998].
  - Terminates when there is no improvement over a current subset.
- Advantages of sequential search
  - computationally advantageous
  - robust against over fitting in producing deterministic results;
  - But may miss optimal subsets.

Multivariate methods - Criterion based selection - AIC

- Assume p variables in total, we have 2^p different potetial models for prediction to evaluate.
- We can evaluate these models with the:
- Akaike Information criterion (AIC): AIC(θ,k) $= -2log\mathcal{L}(\theta) + 2k$
  - k: number of variables, θ: vector of k variables' values, $\mathcal{L}(\theta)$: the probablility to observe the current data given the θ vector.
- *Objective:* select θ, k minimizing AIC (AIC^*: the model with the optimal AIC value)
- k penalizes large models.
- Evaluate the model based on: AIC_k - AIC^*
- For small data sets (n: # data points, $n/k < 40$):

$$\text{AIC}(\theta) = -2log\mathcal{L}(\theta) + 2k + (\frac{2k+1}{n-k-1})\ ^1$$

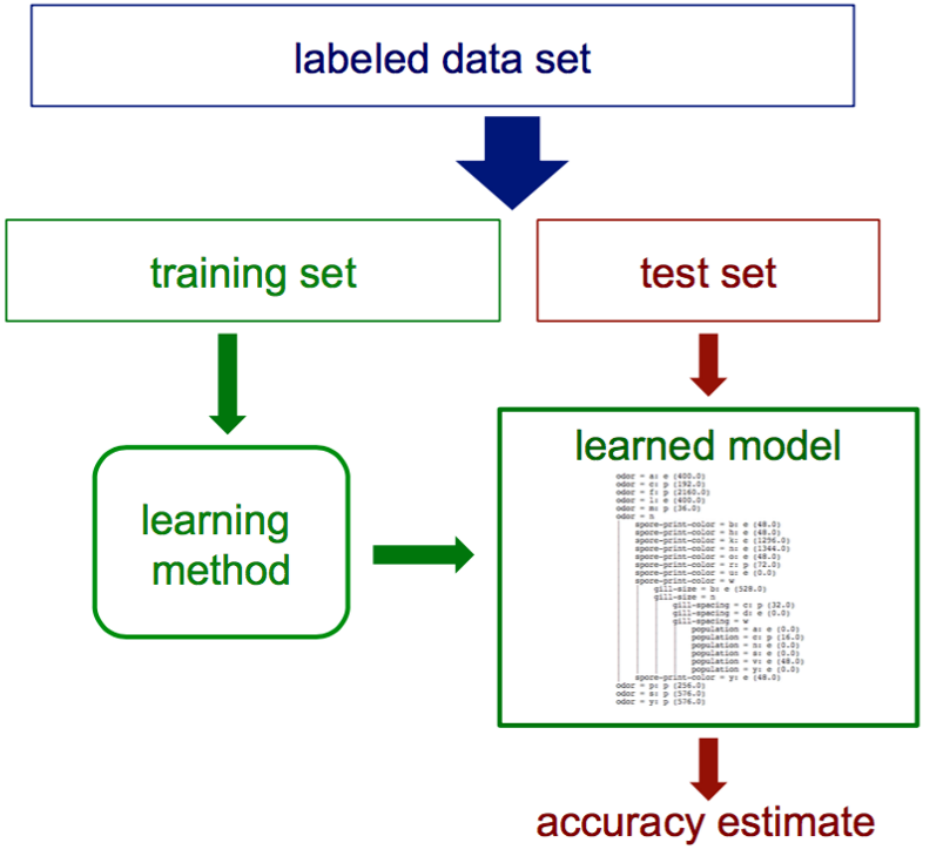
[1] Clifford M. Hurvich and Chih-Ling Tsai, "Regression and Time Series Model Selection in Small Samples," Biometrika 76, no. 2 (June 1989): 297–307.

Multivariate methods - Criterion based selection - BIC

- Assume n data points and p variables in total, we have 2^p different potetial models for prediction to evaluate.
- We can evaluate these models with witht he follwoing two criteria:
- Bayesian Information criterion (BIC): BIC(θ, k) $= -2log\mathcal{L}(\theta) + 2klog(n)$: k: number of variables, *theta*: vector of k variables' values, $\mathcal{L}(\theta)$: the probablility the given the *theta*: vector we observed the current data.
- Objective: select θ, k minimizing BIC (BIC^*: the model with the optimal BIC value)
- k penalizes large models.
- Evaluate the model based on: BIC_k - BIC^* : evidence against a candidate model

Outline

Test sets



Find an unbiased estimate of the accuracy of a learned model?

http://pages.cs.wisc.edu/~dpage/cs760/evaluating.pdf

Learning curves

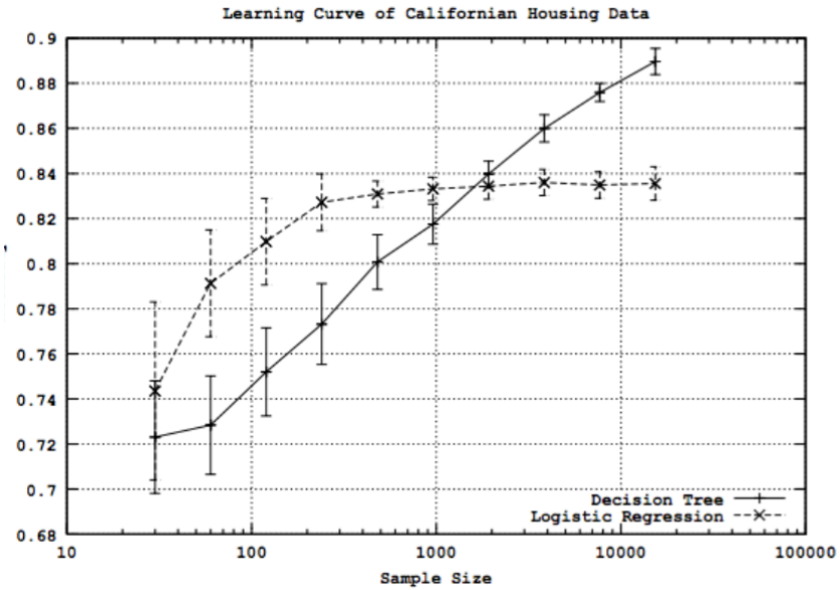


Figure: Figure from Perlich et al. *Journal of Machine Learning Research*, 2003

- Accuracy of prediction vs. training set size.
- **Given training/test set partition**, for each sample size s on learning curve (optionally repeat n times)
  - randomly select s instances from training set
- Learn model
- Evaluate model on test set to determine accuracy a
- Plot (s, avg. accuracy and error bars)

Confusion Matrix

Given training/test set partition

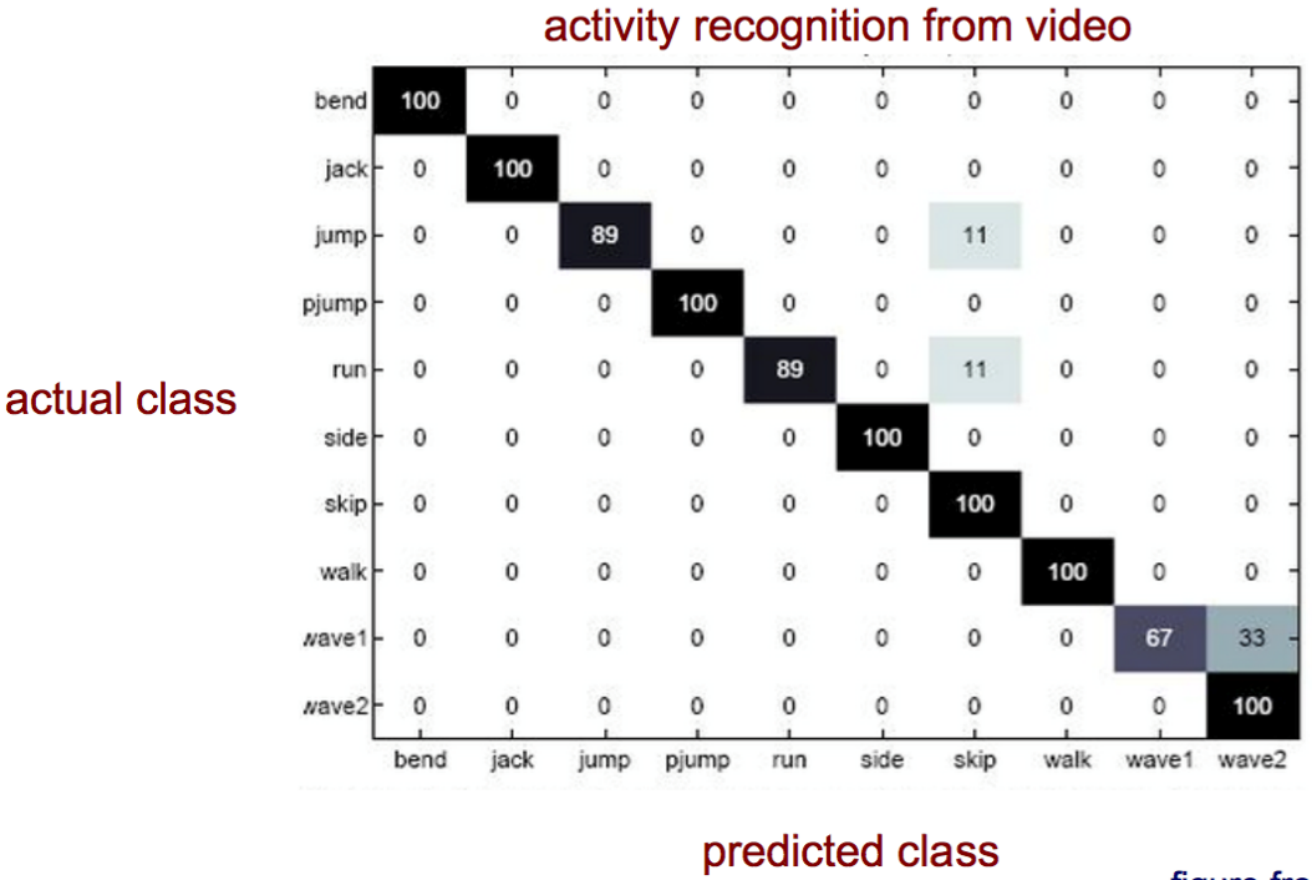
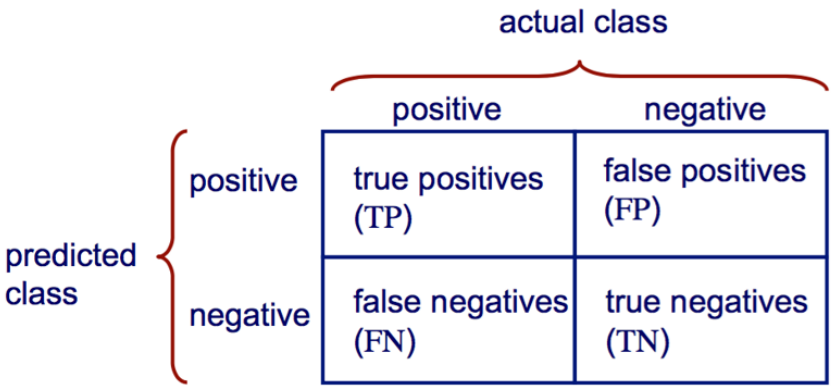


figure from `vision.jhu.edu`

Confusion matrix for 2-class problems



$$\text{accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

$$\text{true positive rate (recall)} = \frac{TP}{\text{actual pos}} = \frac{TP}{TP + FN}$$

$$\text{false positive rate} = \frac{FP}{\text{actual neg}} = \frac{FP}{TN + FP}$$

- Accuracy may not be useful measure in cases where
- There is a large class skew:
  - Is 94% accuracy good if 93% of the instances are negative?
- Cost sensitive classification, getting a positive wrong costs more than getting a negative wrong
  - i. e. In a medical domain false negative results in failure to treat a disease.

ROC curves

- *Receiver Operating Characteristic (ROC)* or **ROC curve**, is a graphical plot illustrating the performance of a binary classifier system as its discrimination threshold (probability) is varied
- Plot true positive rate vs. false positive rate for each possible classification threshold
- Assume classifiers that produce for each data point a probability for positive/negative classification
- ROC visualizes performance for all classification thresholds
- An excellent visualization: `http://www.navan.name/roc/`

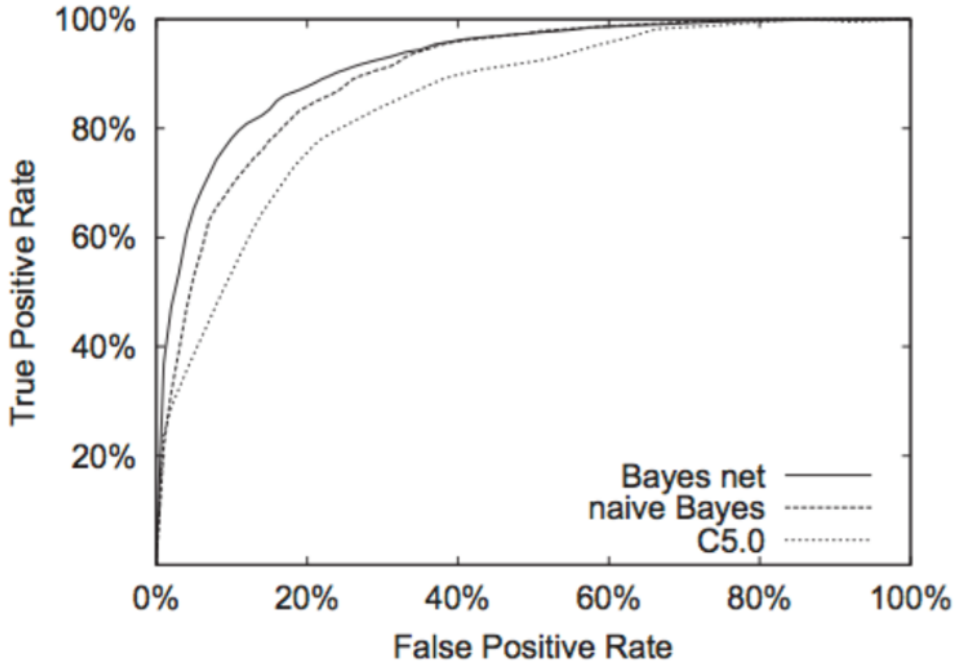


figure from Bockhorst et al., *Bioinformatics* 2003

ROC curves

- For each threshold X in $[v_{min}, v_{max}]$ value range of a variable
  - find FPR_i, TPR_i
  - plot $(\text{FPR}_i, \text{TPR}_i)$
  - recall: $FP_{X>thrs} = p(0|X)$ and $FN_{X>thrs} = p(1) - p(1|X)$
  - Also: $TPR = \frac{TP}{TP+FN}$ and $FPR = \frac{FP}{FP+TN}$

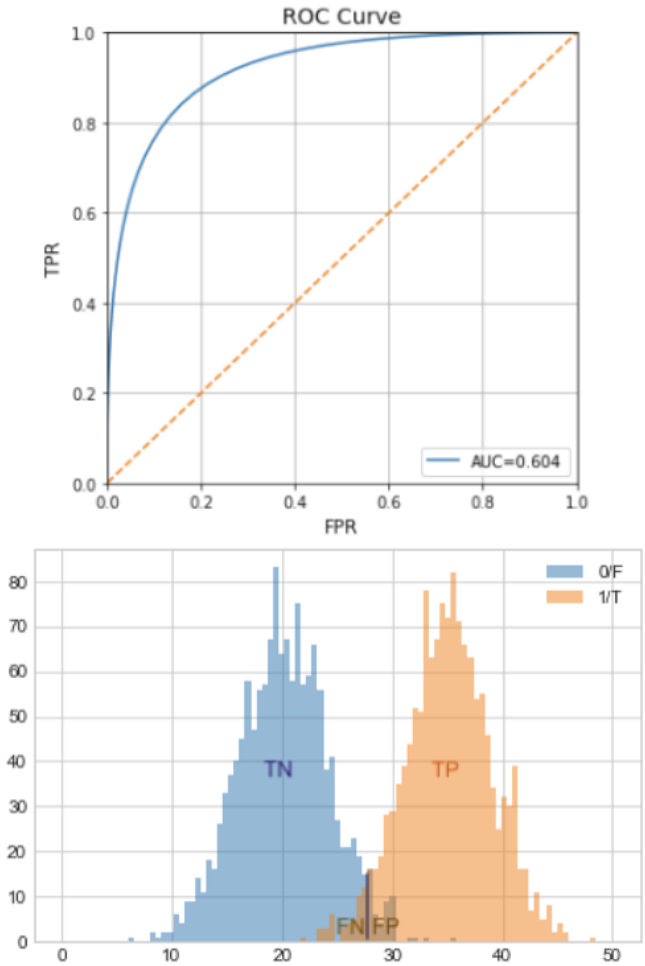
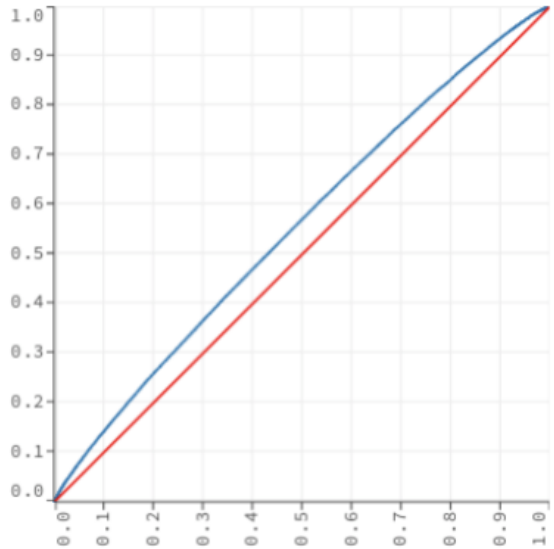
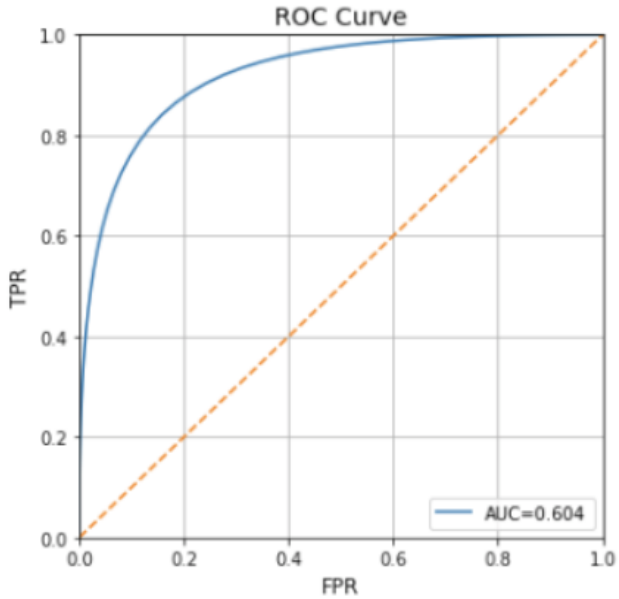
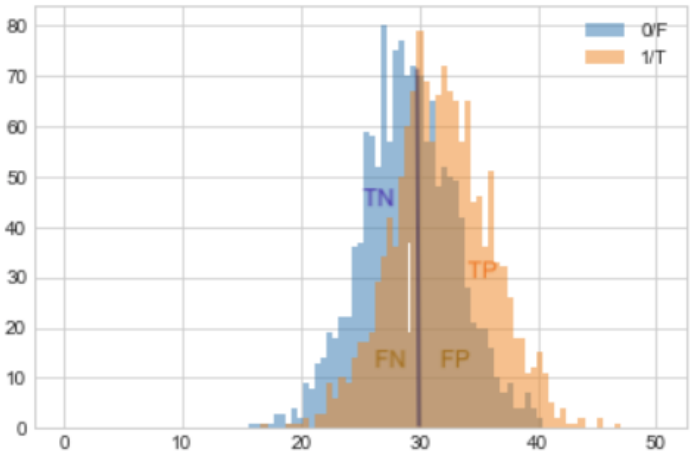


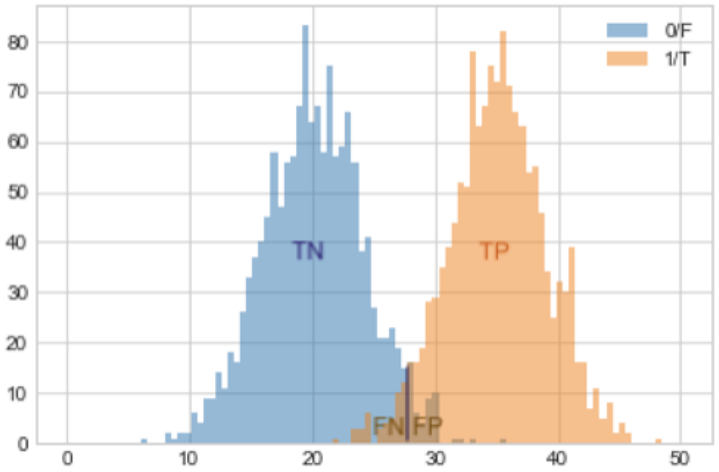
Figure: An excellent visualization: `http://www.navan.name/roc/`

ROC curves


1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0


ROC Curve
TPR
FPR
AUC=0.604
0.0 0.2 0.4 0.6 0.8 1.0


0/F
1/T
TN
TP
FN
FP
0 10 20 30 40 50
0 10 20 30 40 50 60 70 80


0/F
1/T
TN
TP
FN
FP
0 10 20 30 40 50
0 10 20 30 40 50 60 70 80

AUC curve

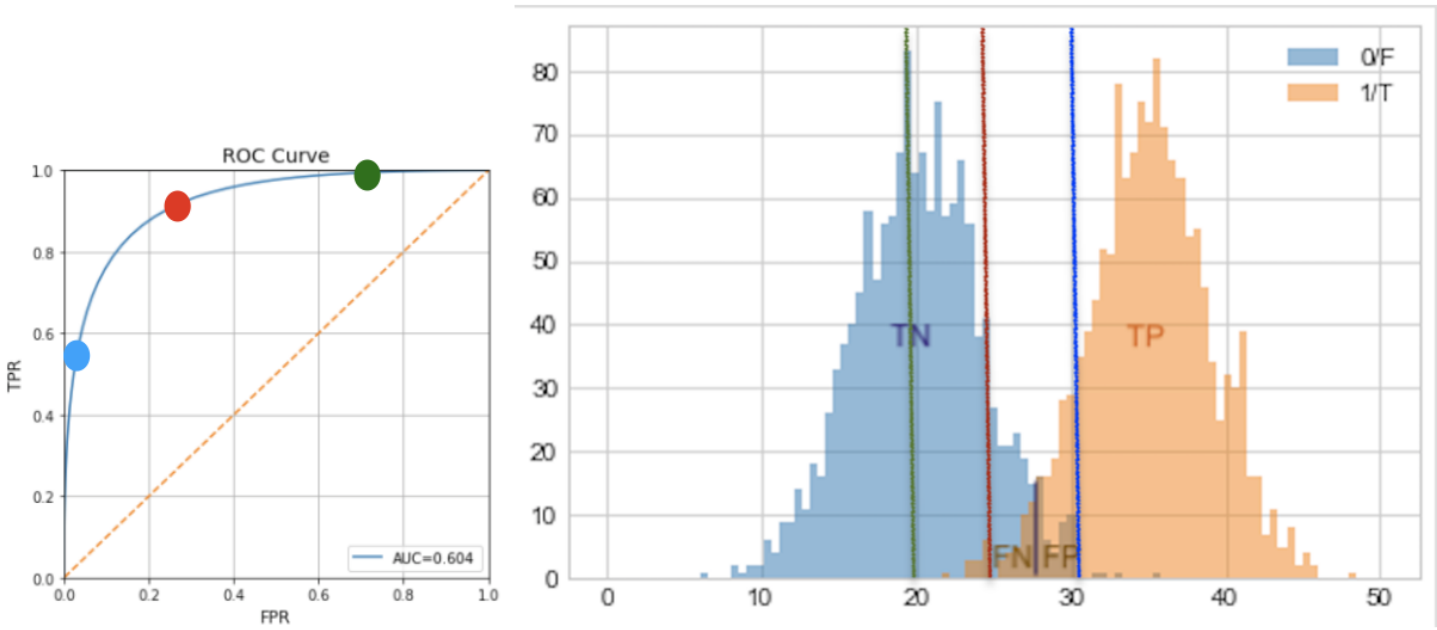


Figure: TPR vs FPR tradeoff based on threshold

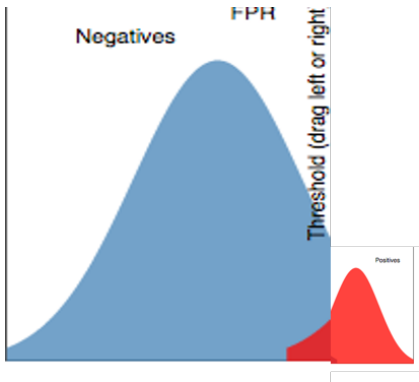
- Quantify the performance of the classifier:
  i.e. thrs $= 20$:

$$TPR = \frac{\sum_{X \geq 20} p(1|X)}{\sum_{X \geq 20} p(1|X) + (p(1) - \sum_{X \geq 20} p(1|X))}$$

$$FPR = \frac{\sum_{X \geq 20} p(0|X)}{\sum_{X \geq 20} p(0|X) + p(0)}$$

ROC curves issues

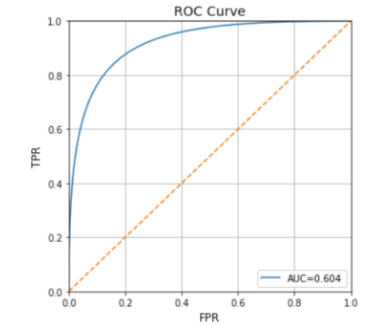
- Not-balanced classes
- Non-normal distributions
- ROC curves robust to non-proper probabilities – only the ranking order counts
  - Good solution for highly unbalanced classes

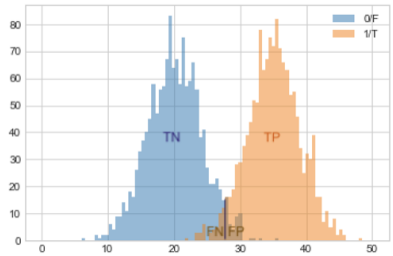


- extended to **classification problems with three or more classes**
  - Class 1 vs Classes 2&3
  - Class 2 vs Classes 1&3
  - Class 3 vs Classes 1&2

ROC curve – Setting the threshold

- how to set your classification threshold, to predict out-of-sample data
- more of a **business decision**,
  - **minimize your False Positive Rate** or
  - **maximize your True Positive Rate**



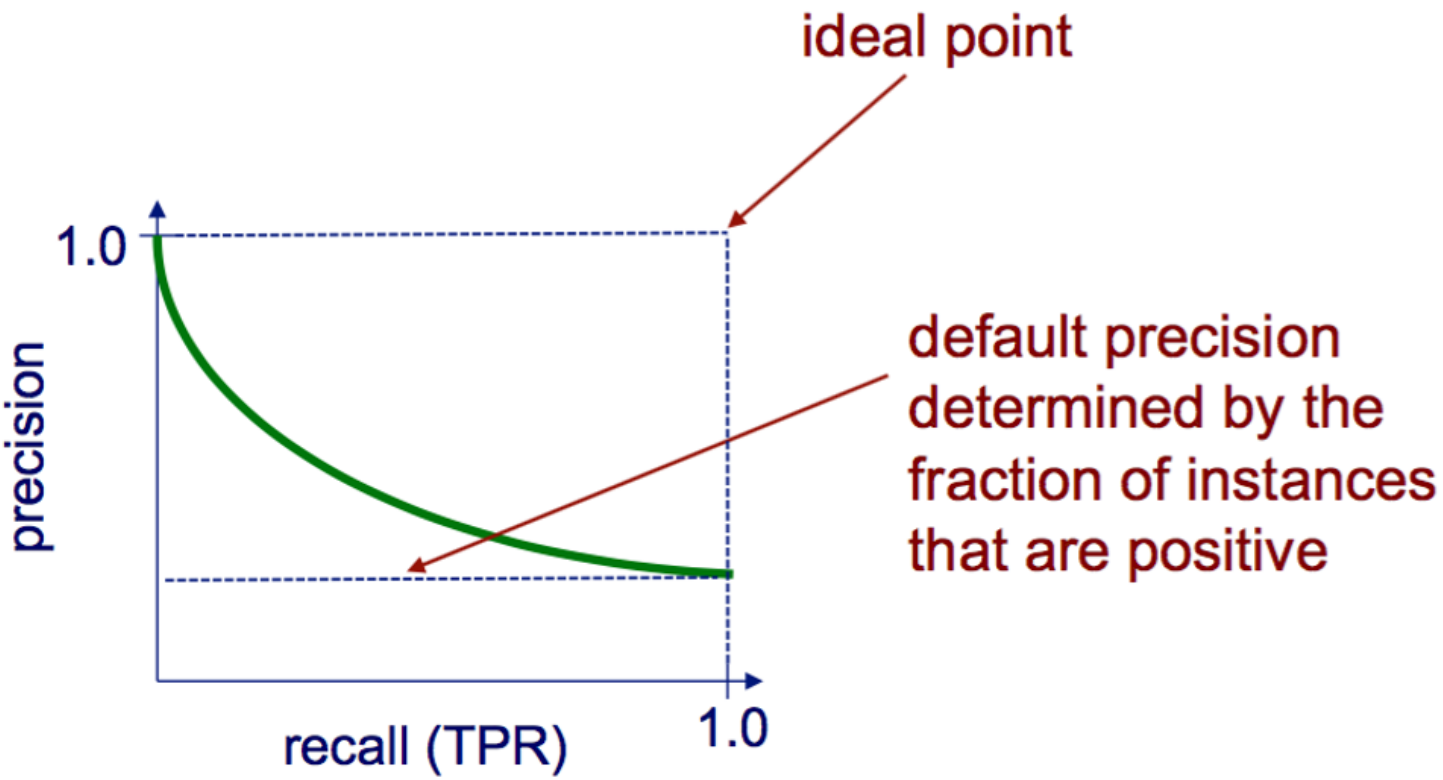


- i. e. classifier to predict credit card transaction might be fraudulent and thus should be reviewed by the credit card holder.
  - business decision set the threshold very low.
  - lot of false positives, but it would maximize the true positive rate.
  - thus minimize the number of cases in which a real instance of fraud was not flagged for review.

Precision – Recall Curves

$$\text{recall (TP rate)} = \frac{\text{TP}}{\text{actual pos}} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{precision} = \frac{\text{TP}}{\text{predicted pos}} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$



PR vs. ROC

Figure: ROC

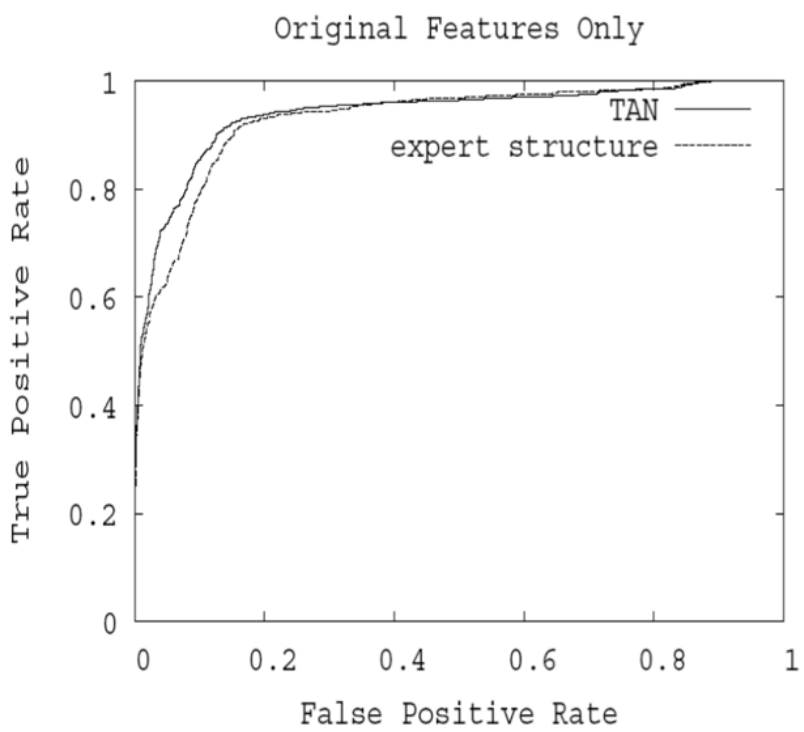
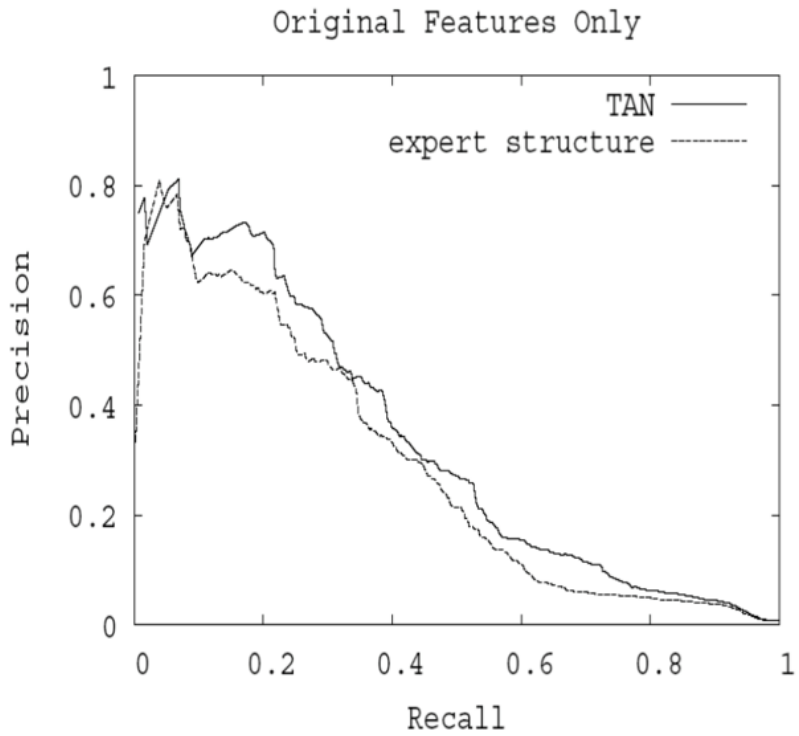


Figure: PR



Outline

VC-Dimension

- Vapnik-Chervonenkis (VC) dimension - pivotal in Learning Theory[2]: theoretical underpinnings of the relationship between #data points and #features in a model is crucial for understanding the performance of machine learning under different conditions.
- VC dimension of a hypothesis (set of models) is a measure of its *capacity* or complexity - the largest number of points that can be shattered (correctly classified) by the hypothesis class.
- VC dimension determines the sample complexity of a learning algorithm: #of training samples required to ensure that the learned model *generalizes well* to unseen data.
- higher VC dimension indicates i. more complex model that can represent more intricate patterns but ii. requires more data to avoid overfitting.

[2]Vapnik, V. N. (1998). Statistical Learning Theory. Wiley-Interscience.

VC-Dimension - #features

- VC dimension increases with the number of features for many hypothesis classes
  - Linear Classifiers: for d-dimensional space, VC dimension of linear classifiers (like linear Support Vector Machines) is $d + 1$: as number of features d increases, the capacity of the model increases linearly.
  - Polynomial Classifiers: VC dimension grows with polynomial degree and the number of features, often leading to much higher capacities.
  - Neural Networks: The VC dimension can be huge -more flexible but data-hungry.
- Sample Complexity
  - theorem relating VC dimension to sample complexity : $m \geq \frac{VC(H)+log(1/\delta)}{\epsilon^2}$
  - m: #training samples, $VC(H)$: VC dimension of hypothesis class H, ϵ: desired generalization error, δ: confidence level- p(generalization error is within ϵ).
- Implications
  - Higher VC Dim (More Features): Requires more data to achieve same generalization .
  - Low VC Dimension: Can achieve good generalization with fewer data points but may lack the capacity to model complex patterns.

VC-Dimension - Generalization bounds

- VC dimension provides bounds on the generalization error: difference between the *training set error* and *expected error on unseen data*.:

$$GeneralizationError \leq TrainingError + \sqrt{\frac{VC(H)log(m)+log(1/\delta)}{m}}$$

- m: #training samples, $VC(H)$: VC dimension of hypothesis class H, ϵ: desired generalization error, δ: confidence level- p(generalization error is within ϵ).
- Interpretation:
  - As VC(H) increases (more features), upper bound on the generalization error increases unless mm also increases.
  - To maintain a low generalization error with more features, the number of data points mm must grow accordingly.

VC-Dimension -Practical Implications for ML Model Credibility

- VC dimension provides bounds on the generalization error: difference between the *training set error* and *expected error on unseen data*.:

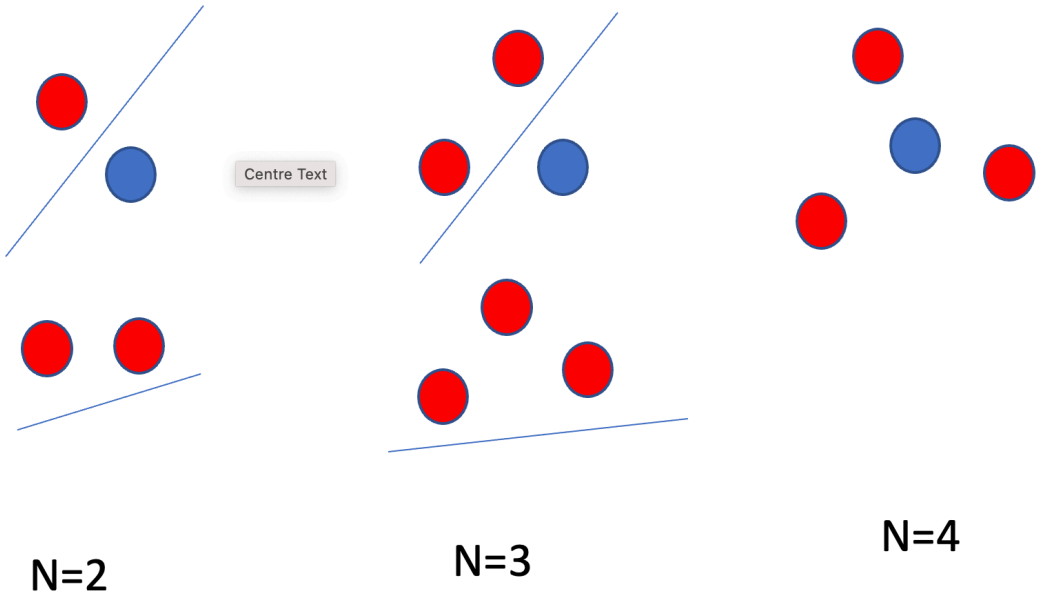
$$GeneralizationError \leq TrainingError + \sqrt{\frac{VC(H)log(m)+log(1/\delta)}{m}}$$

- m: #training samples, $VC(H)$: VC dimension of hypothesis class H, ϵ: desired generalization error, δ: confidence level- p(generalization error is within ϵ).
- Interpretation:
  - As VC(H) increases (more features), upper bound on the generalization error increases unless mm also increases.
  - To maintain a low generalization error with more features, the number of data points mm must grow accordingly.

VC-Dimension -an Example

- Hypothesis Class (H): All possible linear classifiers (lines) in a 2D plane: $ax + by + c = 0$, a, b, c: real numbers for line's slope and position.
- determine VC dimension:
  - Shattering a Set of Points: A hypothesis class H shatters a set of points if, for every possible labeling (assignment of classes) of the points, there exists at least one hypothesis in the class that perfectly separates the points.
  - Finding the Largest N Shattered by H: find maximum number of points N such that H can shatter any set of N points in general positions (no three points are collinear).

VC-Dimension - an Example



Final Determination: VC dimension of H is 3.

VC-Dimension -an Example

- We want to ensure that our linear classifier has a generalization error (ϵ) of at most 5% (0.05) with a confidence level $(1-\delta)$ of 95% ($\delta = 0.05$). Determine the minimum number of training samples (m) required to achieve this.
- Based on: $m \geq \frac{VC(H)+log(1/\delta)}{\epsilon^2}$
  for $VC(H) = 3$, ϵ: 5% , $(1-\delta)$ of 95% ($\delta = 0.05$)
- $m \geq 2398.28$: need at least 2,399 training samples to ensure that a linear classifier in 2D achieves a generalization error of at most 5% with 95% confidence.

References I

Christopher M. Bishop

Pattern Recognition and Machine Learning

2006

Trevor Hastie, Robert Tibshirani, Jerome Friedman

The Elements of Statistical Learning: Data Mining, Inference, and Prediction

Second Edition. 2009

Shai Shalev-Shwartz, Shai Ben-David

Understanding Machine Learning: theory and algorithms

Cambridge University Press. 2014

Robert Tibshirani

Regression shrinkage and selection via the Lasso

Journal of the Royal Statistics Society. 58(1), 267-288. 1996

`http://statweb.stanford.edu/~tibs/lasso/lasso.pdf`

`http://people.inf.elte.hu/kiss/13dwhdm/roc.pdf`

References II

Hui Zou, Trevor Hastie

Regularization and Variable Selection via the Elastic Net

Journal of the Royal Statistical Society. Series B (Statistical Methodology). Wiley.67(2): 301–20. 2005

Jerome Friedman, Trevor Hastie, Robert Tibshirani

A note on the group lasso and a sparse group lasso

`https://arxiv.org/pdf/1001.0736.pdf`

[Guyon and Elisseeff, 2003]p8 Isabelle Guyon, Andre Elisseeff,

An Introduction to Variable and Feature Selection

Journal of Machine Learning Research. 3 (2003) 1157-1182