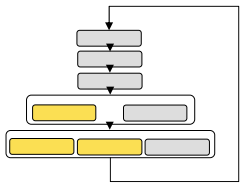
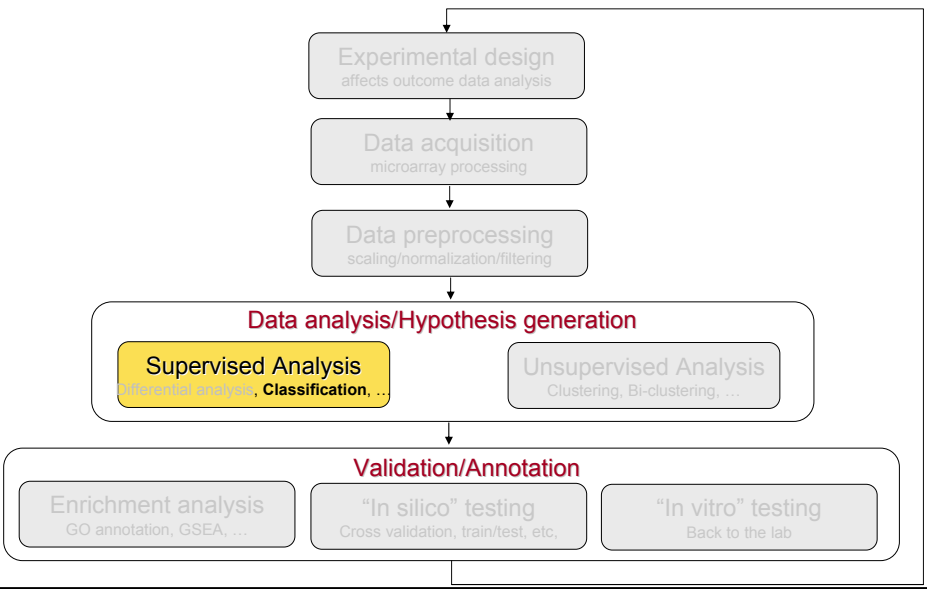
Class Prediction

Classification



The functional genomics pipeline



Classification

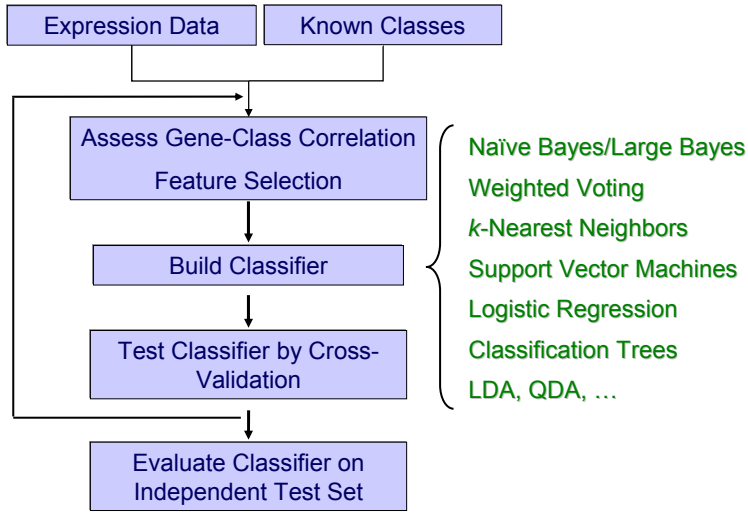
Given phenotypically distinct classes, find a gene expression signature that accurately predicts class membership.

Take-home messages

- How to build a classifier.
- How to evaluate a classifier.
- What to evaluate of a classifier.

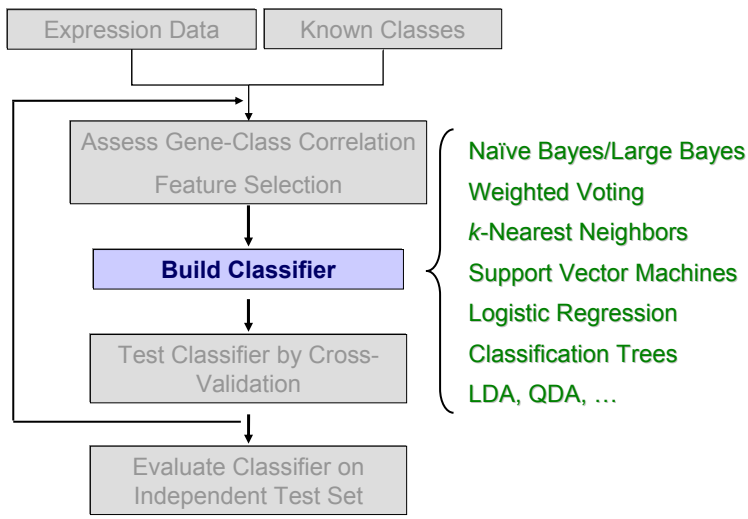
Classification

Computational Methodology



Classification

Computational Methodology



Classifiers

- Too many to discuss
  - naïve Bayes, Large Bayes …
  - KNN, shrunken centroids, association rules ..
  - Neural Networks, SVM, logistic regression ..
  - …
- Important issues:
  - small *N*, large *p*: few cases, many variables (genes).
  - redundancy: many highly correlated genes.
  - noise: measurements are very imprecise.
  - feature selection: reducing *p* is a necessity.
  - evaluation: avoid over-fitting.

Probabilistic Classifiers

- Estimate the probability

$$P(\mathbf{G} \mid C), \text{ where } \mathbf{G} = \{g_1, g_2, ..., g_m\}$$

- Bayes Theorem:

$$P(C \mid \mathbf{G}) = \frac{P(\mathbf{G} \mid C)P(C)}{P(\mathbf{G})} = \frac{P(\mathbf{G} \mid C)P(C)}{\sum_{C'} P(\mathbf{G} \mid C)P(C')}$$

$$\propto P(\mathbf{G} \mid C)P(C)$$

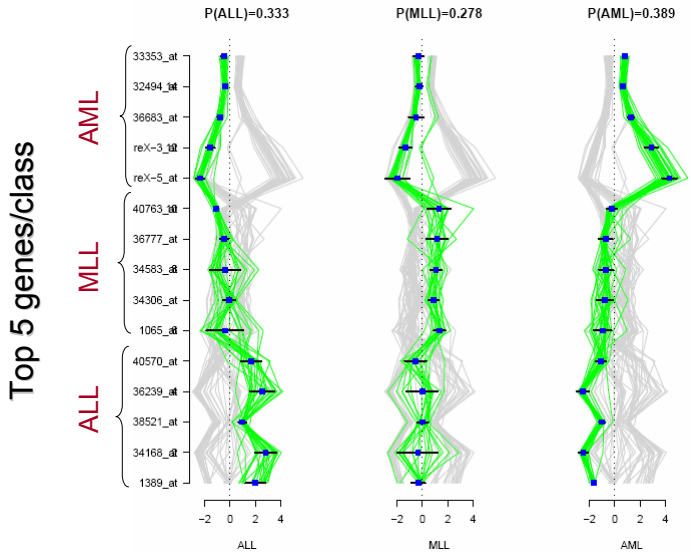
- Classification:

$$\text{Class}(\mathbf{G}) = \arg\max_{C}[P(C \mid \mathbf{G})]$$

WARNING: assumption of equal misclassification costs

Naïve Bayes

class centroids based on within-class mean and stdev



Proabilistic classification

naïve-Bayes

- Assumption:
  - genes conditionally independent given the class variable.

$$P(\mathbf{G}\,|\,C) = P(g_1\,|\,C) \times P(g_2\,|\,C) \times \ldots P(g_m\,|\,C)$$

$$= \prod_{i=1}^{m} P(g_i\,|\,C)$$

$$P(C\,|\,\mathbf{G}) = \frac{P(C)P(\mathbf{G}\,|\,C)}{P(\mathbf{G})} \propto P(C)\prod_i P(g_i\,|\,C)$$

Proabilistic classification
naïve-Bayes

- Assumption:
  - genes conditionally independent given the class variable.

$$P(\mathbf{G} \mid C) = P(g_1 \mid C) \times P(g_2$$
$$= \prod_{i=1}^{m} P(g_i \mid C)$$

$$P(g_i \mid C = k) \sim N(\mu_{ik}, \sigma_{ik}^2)$$
$$\hat{\mu}_{ik} = \frac{1}{n_k} \sum_{j:c_j = k} g_{ij}$$
$$\sigma_{ik}^2 = \frac{1}{n_k - 1} \sum_{j:c_j = k} (g_{ij} - \hat{\mu}_{ik})^2$$

$$P(C \mid \mathbf{G}$$

$P(C = k) = \frac{n_k}{n}$ ML estimator

$P(C = k) = \frac{n_k + 1}{n + 2}$ Bayes estimator

$$P(C) \prod_i P(g_i \mid C)$$

Proabilistic classification
naïve-Bayes

- Assumption:
  - genes conditionally independent given the class variable.

$$P(\mathbf{G} \mid C) = P(g_1 \mid C) \times P(g_2 \mid C) \times \ldots P(g_m \mid C)$$
$$= \prod_{i=1}^{m} P(g_i \mid C)$$

$$\log P(C \mid \mathbf{G}) \propto \log P(C) + \underbrace{\log P(g_1 \mid C)}_{\text{Gene } g_1\text{'s vote}} + \ldots + \underbrace{\log P(g_p \mid C)}_{\text{Gene } g_p\text{'s vote}}$$

$-\frac{(g_1 - \mu_{1C})^2}{\sigma_{1C}^2}$

Linear Discriminant Analysis

- Generalization of naïve-Bayes

$$P(\mathbf{G} \mid C = k) \sim N(\mu_k, \Sigma_k)$$

Multivariate Gaussian
- w/ non-diagonal covariance matrix
- can be:
  - ✓ class dependent $\Rightarrow \Sigma_k$
  - ✓ class independent $\Rightarrow \Sigma$

$$\mu_k = (\mu_{k1}, \mu_{k2}, \ldots, \mu_{kp})$$

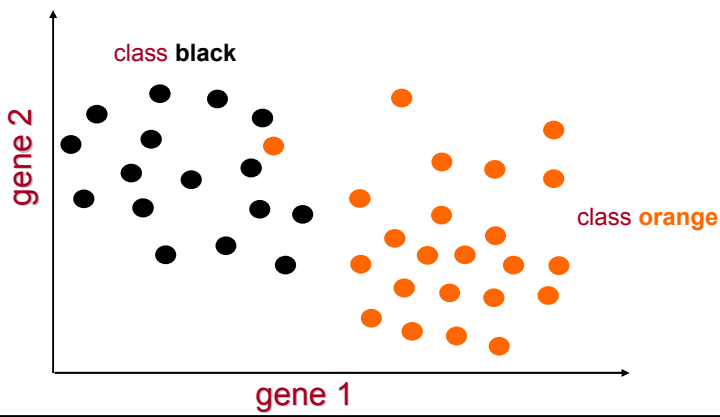
$$\Sigma_k = \begin{bmatrix} \sigma_{11}^k & \sigma_{12}^k & .. & \sigma_{1p}^k \\ \sigma_{21}^k & \sigma_{22}^k & .. & \sigma_{2p}^k \\ & & .. & \\ \sigma_{p1}^k & \sigma_{p2}^k & .. & \sigma_{pp}^k \end{bmatrix}$$

- Low-dimension needed
- Often combined w/ PCA, SVD, etc.

K-nn classifier

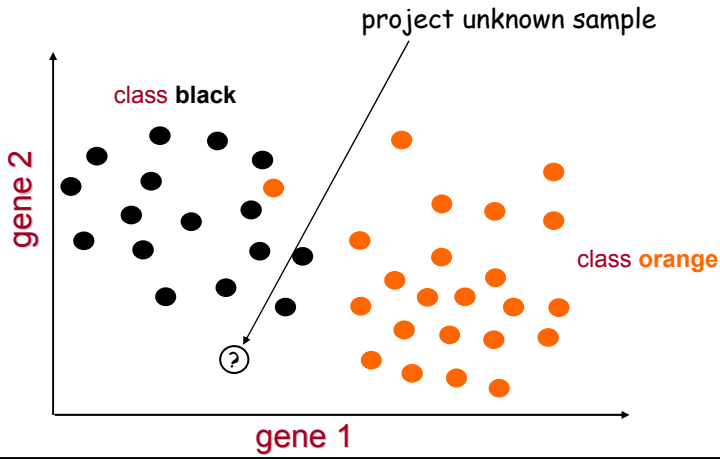
example: K=5, 2 genes, 2 classes

project samples in gene space



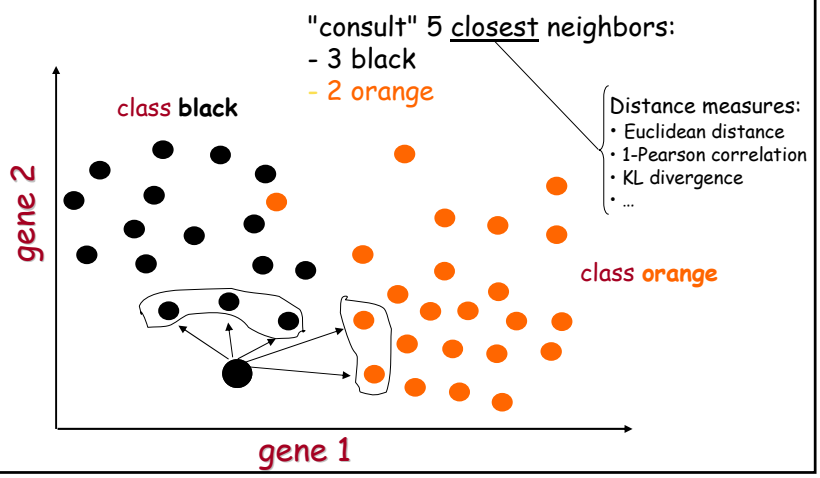
K-nn classifier

example: K=5, 2 genes, 2 classes



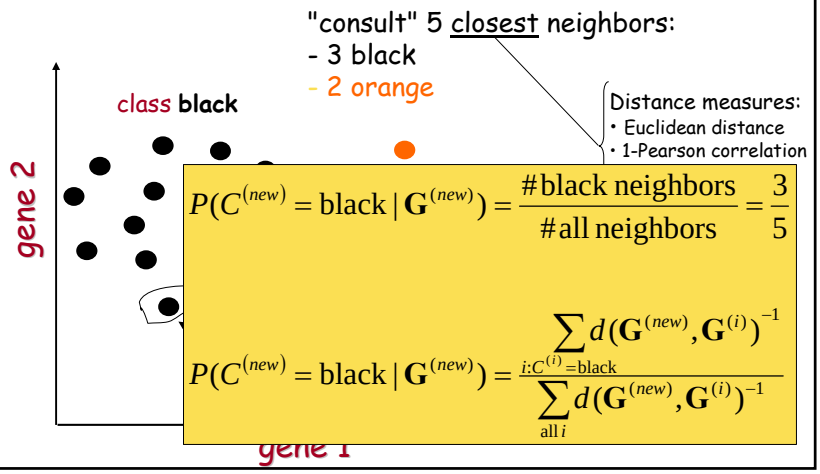
K-nn classifier

example: K=5, 2 genes, 2 classes



K-nn classifier

example: K=5, 2 genes, 2 classes



$$P(C^{(new)} = \text{black} \mid \mathbf{G}^{(new)}) = \frac{\#\text{black neighbors}}{\#\text{all neighbors}} = \frac{3}{5}$$

$$P(C^{(new)} = \text{black} \mid \mathbf{G}^{(new)}) = \frac{\sum_{i:C^{(i)}=\text{black}} d(\mathbf{G}^{(new)}, \mathbf{G}^{(i)})^{-1}}{\sum_{\text{all } i} d(\mathbf{G}^{(new)}, \mathbf{G}^{(i)})^{-1}}$$

Probabilistic NN (PNN)

- PNN's output is the posterior probability $P(c|\mathbf{x})$.

Class prior prob.

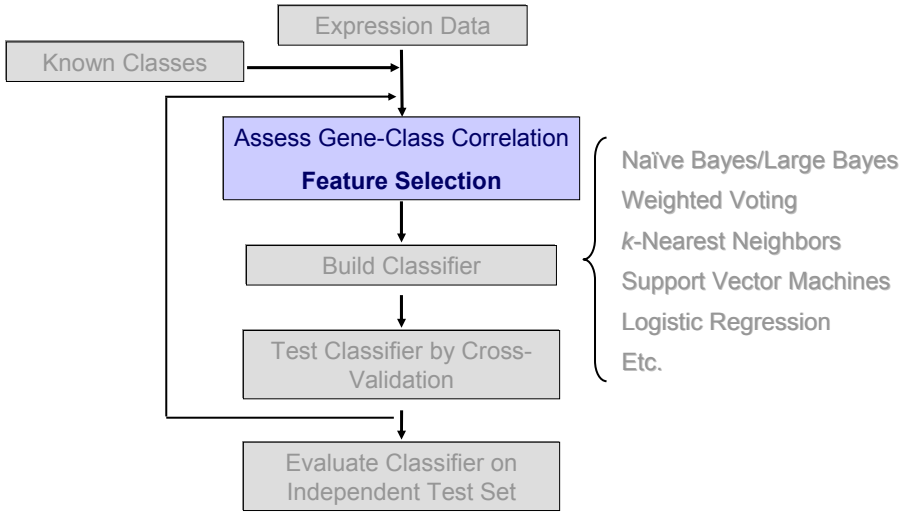
$$P(c \mid \mathbf{G}^{new}) = \frac{P(\mathbf{G}^{new} \mid c)P(c)}{\sum_{c'} P(\mathbf{G}^{new} \mid c')P(c')} = \frac{\frac{P(c)}{n_c}\sum_{i:\mathbf{G}_i \in c} \exp\left(-\mathrm{D}(\mathbf{G}^{new}, \mathbf{G}_i)^2 / 2\sigma^2\right)}{\sum_{c'}\left[\frac{P(c')}{n_{c'}}\sum_{i:\mathbf{G}_i \in c'} \exp\left(-\mathrm{D}(\mathbf{G}^{new}, \mathbf{G}_i)^2 / 2\sigma^2\right)\right]}$$

Bayes Rule

Estimate density $P(\mathbf{G}|c)$ by placing a Gaussian on each of the training set samples

- σ acts as k in kNN, when σ is small it is like kNN with k=1.
- D(**x**,**y**) = Euclidean distance in the space of selected features.
- *P(c) = 1/n* to belong to a class, *(n-1)/n* not to belong to a class, where n is the number of classes.

Classification
Computational Methodology



Classification
feature/gene selection

- Motivation
  - Large number of genes, with many likely to contain no signal.
  - Use of all genes may lead to over-fitting
  - Practical: makes the model building faster.
- Methods
  - Univariate methods: consider genes/features individually
  - Multivariate methods: take into account dependence among features.

Classification
univariate feature/gene selection

- Univariate feature selection:
  - Same as for differential analysis.
    - Rank genes by their "correlation" with phenotype (by t-score, etc.)
    - Select top *n* ranked genes per class.
  - Need to use 2-level CV for "fair" error estimation.

- *Cons*
  - *Cannot capture feature interaction*
  - *Selection of redundant features (highly correlated).*

- *Pros*
  - *Selection of redundant features (highly correlated).*
  - *Simple and fast.*

Classification
multivariate feature/gene selection

- Step-wise feature selection (forward/backward)

  Begin (input: dataset $\mathbf{D}_{m,n+1}$)
  - Feature set $\mathbf{G}=\emptyset$ (or $\mathbf{G}_{1:n}$), predictive score S(D,**G**)
  - Repeat:
    - Find feature $g_{best} = \text{argmax}_g\ [S(D, g \cup \mathbf{G})]$
    - Add feature: $\mathbf{G}=g_{best} \cup \mathbf{G}$

    } Forward step

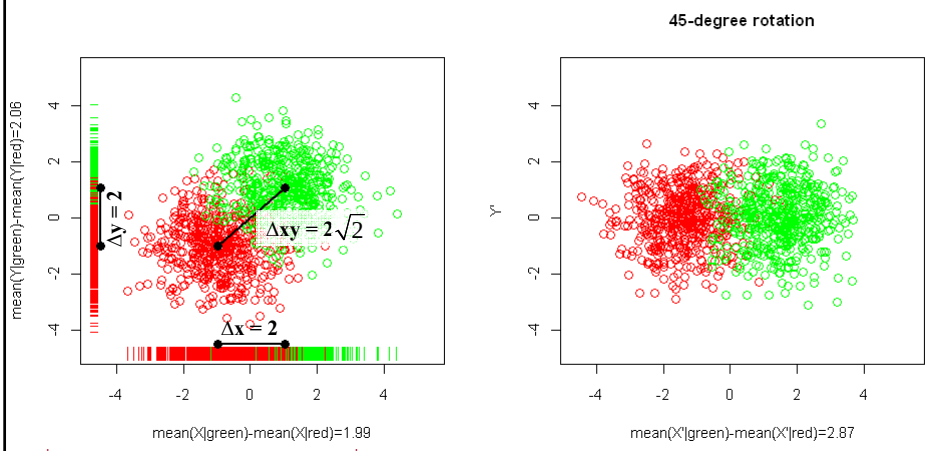
    - Find feature $g_{best} = \text{argmax}_g\ [S(D, \mathbf{G} - g)]$
    - Delete feature: $\mathbf{G}=\mathbf{G} - g_{best}$

    } Backward step
  - Until no more improvement

  End

- *Pros/Cons complementary to 1st approach.*
  - *no redundant features; computationally intensive*

Good *vs.* Bad Redundancy




```
X|red ~ N(-1,1)     cor(X,Y|red)≅   0
Y|red ~ N(-1,1)     cor(X,Y|green)≅ 0
X|green ~ N(1,1)    cor(X,Y)=       0.45
Y|green ~ N(1,1)
```

Features are not really redundant when correlation is induced by class membership.

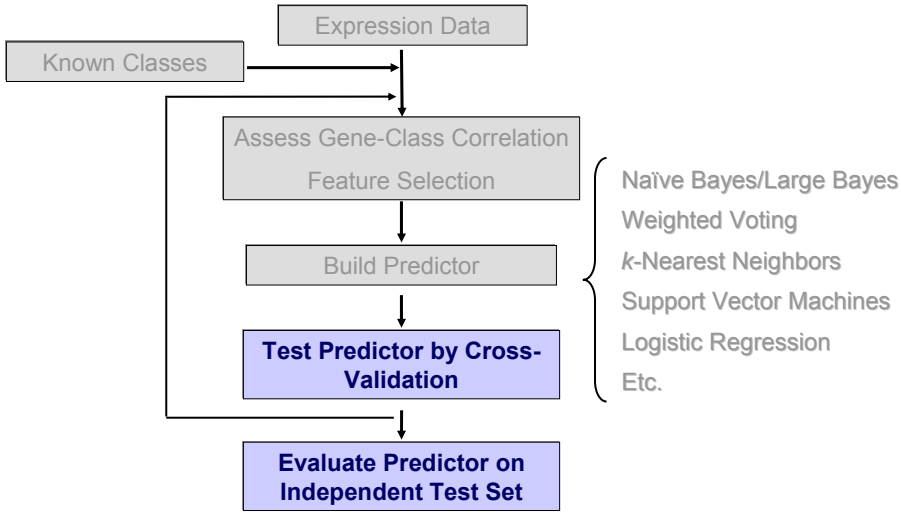
Classification

feature/gene selection

- Univariate: select top *n* genes/class (by t, SNR, ...)
  - Same as for differential analysis.
- Step-wise feature selection (greedy search)
  - Feature set **G**=Ø, predictive score S(D,**G**)
  - Repeat:
    - Find feature g_{best} = $argmax_g$ [S(D,g∪**G**)]
    - Add feature: **G**=g_{best}∪**G**
  - Until no more improvement
- Others:
  - Simulated annealings, GA, combination schemes.

Class Prediction

Computational Methodology



Testing the classifier

- Evaluation on independent test set
  - Build the classifier on the train set.
  - Assess prediction performance on test set.
    - Maximize generalization/Avoid overfitting.
- Performace measures …

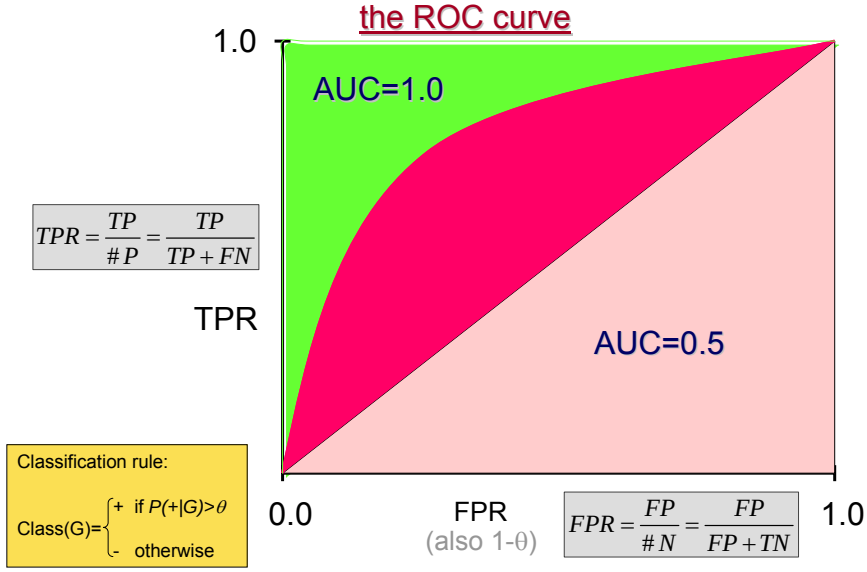
How to assess a classifier

performance measures

- Error rate = $\frac{\text{\# of cases correctly classified}}{\text{total \# of cases}}$
- ROC curve
  - plots false positive rate (FPR) against true positive rate (TPR) as the threshold for positive classification is varied from 1 to 0.
- Log score
  - $LS = \sum_{i=1}^{n} \log P(c_i \mid G_i)$
- Breier statistic, etc.

Performance Measures

the ROC curve



$TPR = \frac{TP}{\#P} = \frac{TP}{TP + FN}$

Classification rule:

Class(G)= { + if $P(+|G) > \theta$; - otherwise

$FPR = \frac{FP}{\#N} = \frac{FP}{FP + TN}$

Testing the classifier

- Evaluation on independent test set
  - What if we don't have an independent test set?
- Cross Validation (CV):
  - Split the dataset into n folds (e.g., 10 folds of 10 cases each).
  - For each fold (e.g., for each group of 10 cases),
    - train (i.e., build model) on n-1 folds (e.g., on 90 cases),
    - test (i.e., predict) on left-out fold (e.g., on remaining 10 cases).
  - Combine test results.
  - Usually leave-one-out CV (small sample size).

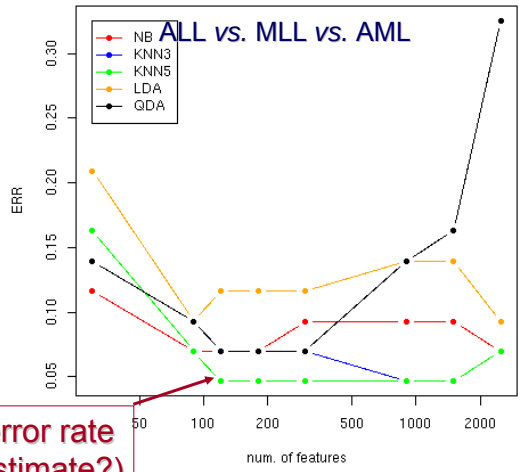
Important: all steps of model-building must be brought into the CV loop (feature selection, discretization, etc.).

Testing the classifier

- Evaluation on independent test set
  - What if we don't have an independent test set?
- Cross Validation (CV):
  - Split the dataset into n folds (e.g., 10 folds of 10 cases each).
  - For each fold (e.g., for each group of 10 cases),
    - train (i.e., build model) on n-1 folds (e.g., on 90 cases),
    - test (i.e., predict) on left-out fold (e.g., on remaining 10 cases).
  - Combine test results.
  - Usually leave-one-out CV (small sample size).
- Also used for Model Selection (choose the best model/best parameters).

Testing the classifier

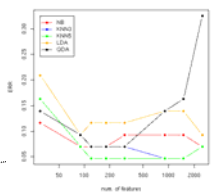
learning curves (LOOCV)



Best error rate (fair estimate?)

Is the error rate significant?

permutation test

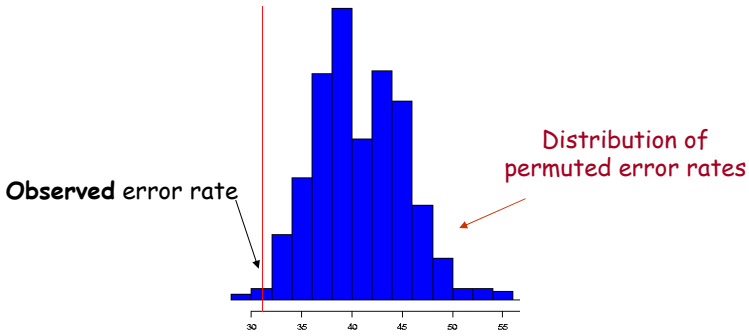


- Permutation test:
  - Shuffle the class labels.
  - Compute error rate on shuffled labels by CV.
  - Repeat many times.
- Empirical distribution on error rate.

Important: all steps of model-building must be brought into the permutation loop (feature selection, etc.)

Testing the classifier

permutation test – empirical null distribution



$$p = \frac{\#\{\text{err}_{permuted} \leq \text{err}_{observed}\}}{\#\{\text{err}_{permuted}\}}$$

Classification

error rate: significance *vs*. error estimate

- Permutation test not appropriate to estimate probability of correct prediction on future cases.
- Use of independent test set is best.
- Two-level Cross-Validation (CV^2).
  - error bar estimation problematic.

Testing the classifier

recap

- Error rate estimate:
  - Evaluate on independent test set:
    - Best error estimate and error bar.
  - "Resampling" methods
    - Needed when small sample size and for model selection.
    - Cross Validation (CV)
    - Bootstrapping
    - Random splits
- Significance:
  - asymptotic p-value.
  - permutation test.

How to assess a classifier

performance measures

Classification accuracy

- Error rate = $\frac{\text{\# of cases correctly classified}}{\text{total \# of cases}}$
- ROC curve
  - plots false positive rate (FPR) against true positive rate (TPR) as the threshold for positive classification is varied from 1 to 0.

Calibration

- Log score
  - $LS = \sum_{i=1}^{n} \log P(c_i \mid G_i)$
- Breier statistics, etc.

Probabilistic Classifiers

- Estimate the probability

$$P(\mathbf{G} \mid C), \text{ where } \mathbf{G} = \{g_1, g_2, ..., g_m\}$$

- Bayes Theorem:

$$P(C \mid \mathbf{G}) = \frac{P(\mathbf{G} \mid C)P(C)}{P(\mathbf{G})} = \frac{P(\mathbf{G} \mid C)P(C)}{\sum_{C'} P(\mathbf{G} \mid C)P(C')}$$

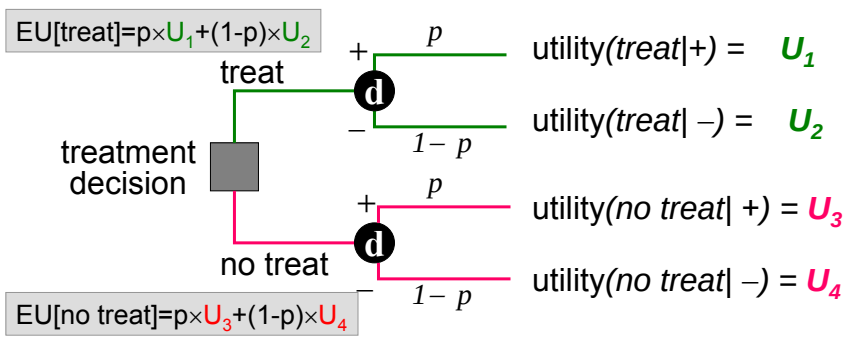
$$\propto P(\mathbf{G} \mid C)P(C)$$

- Classification:

$$\text{Class}(\mathbf{G}) = \begin{cases} \text{cancer} & \text{if } P(\text{cancer} \mid \mathbf{G}) > 0.5 \\ \text{normal} & \text{otherwise} \end{cases}$$

Calibration

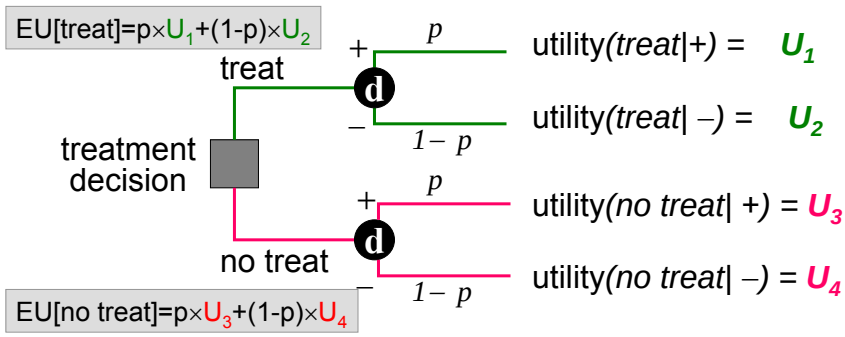
is needed to make optimal decisions



Optimal decision: treat if *EU[treat]>EU[no treat]*

Calibration

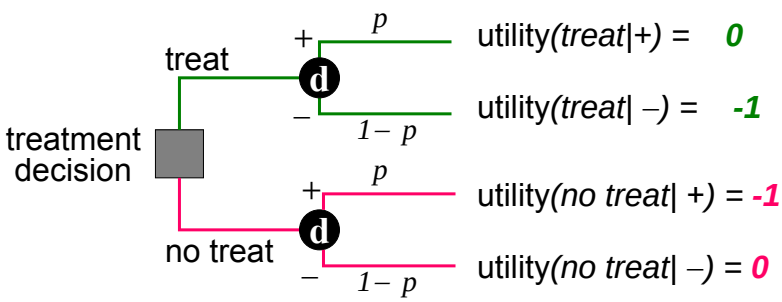
is needed to make optimal decisions



Optimal decision: treat if $p > \frac{U_4 - U_2}{U_4 - U_2 + U_1 - U_3}$

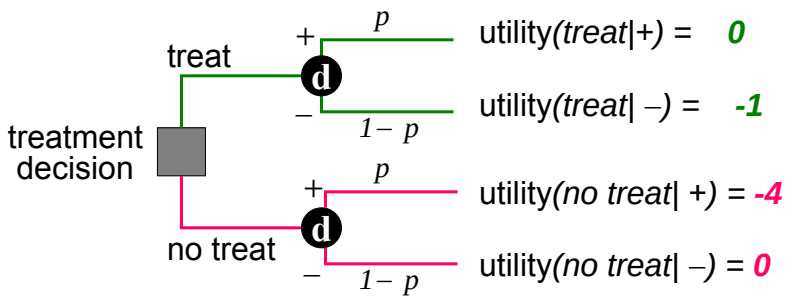
Optimal decision

under zero-one loss



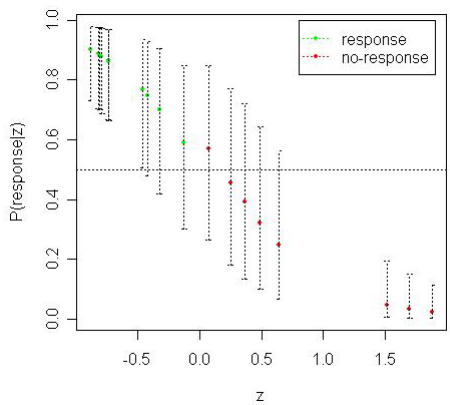
Optimal decision: treat if $p > \frac{1}{2} = 0.5$

Optimal decision
under a-symmetric loss



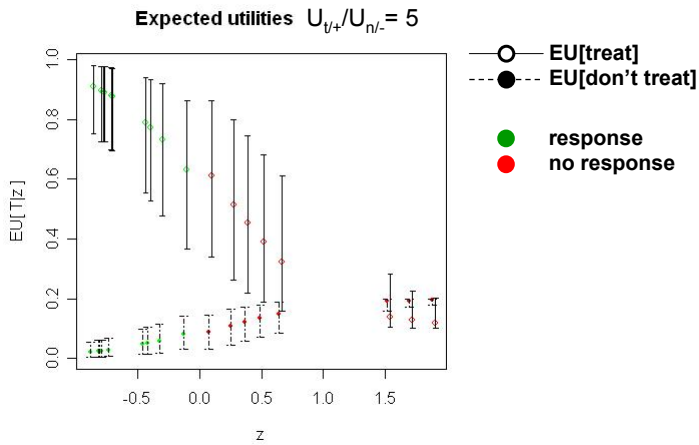
Optimal decision: treat if $p > \frac{1}{5} = 0.2$

An example
predicting drug response



An example

treatment decision based on prediction of drug response



Classification

Take-home messages

- How to build a classifier.
  - Feature selection usually beneficial (use "good" redundancy).
  - Probabilistic classifiers desirable: naïve-Bayes, LDA, PNN, etc.
- How to evaluate a classifier.
  - Evaluation on independent test data
  - Cross-Validation (for model selection too)
  - Significance by permutation test
- What to evaluate of a classifier.
  - Error rate
  - ROC curve
  - Log-score
  - …

Classification
Cookbook

- Start by splitting data into train and test set (stratified). "Forget" about the test set until the very end.
- Explore different feature selection methods and different classifiers on train set by CV.
  - Make sure to bring ALL model building choices within CV loop.
- Once the "best" classifier and best classifier parameters have been selected (based on CV)
  - Build a classifier with given parameters on entire train set.
  - Apply classifier to train set.
- When evaluating the classifier performance, take into account the decision for which the classification is needed.

WARNING: should not be used blindly …

Classification
not discussed

- Other classifiers:
  - Weighted voting
  - (logistic) regression
  - SVM, ANN, BN
  - "Ensemble classifiers" (combining classifiers):
    - Bagging, Boosting, Random Forests, others.
- Bootstrapping, Random Resampling, others.
- Cross-platform classification …

Cross-platform analysis

- Often, the classifier built on a dataset needs to be applied to a different datasets.
- Different datasets may incorporate significant biases that make the comparison difficult:
  - Different chips (e.g., U95 *vs.* U133)
  - different platforms (e.g., cDNA vs. oligo)
  - different operators, processing, environmental conditions, etc.
- Some variable normalization or transformation is needed:
  - Sample (column) normalization.
  - Replace values with ranks.
  - Define new variables (i.e., gene ratios: g_i/g_j, combinatorial explosion)
  - Use "robust" classification methods.
- Main challenge to deployment of technology in clinical settings

The End