Classification with microarray data

Aron C. Eklund
eklund@cbs.dtu.dk

DNA Microarray Analysis
January 11, 2013
1. The concepts of classification and prediction
   and how they relate to microarrays and biological/medical problems

2. Machine learning methods for deriving classifiers
   LDA, kNN, SVM, etc.

3. Assessment of classifier performance
   Accuracy, specificity, sensitivity, survival analysis, ROC curves, ...

4. Other considerations
   Feature selection, overfitting, validation and cross-validation.
Track A: The DNA Array Analysis Pipeline

Question/hypothesis

Experimental Design

Design Array or buy a standard array

Sample Preparation Hybridization

NGS data

Image analysis

Mapping

Normalization

Comparable Gene Expression Data

Statistical Analysis Fit to Model (time series)

Advanced Data Analysis
- Clustering
- PCA
- Gene Annotation Analysis
- Promoter Analysis
- Classification
- Meta analysis
- Survival analysis
- Regulatory Network

Friday, January 11, 2013
What is a classifier?

... an algorithm or rule:

Input: (Multivariate) data from an individual

Classifier

Output: “Class” of the individual

Expression profile of a tumor

aggressive or benign

Meteorological measurements

rain / no rain

Digitized fingerprint scan

authorized or unauthorized

Useful when:

• Class is difficult to observe directly
• Class is a prediction about the future
Classification with gene expression profiles

expression profile of a tumor specimen

<table>
<thead>
<tr>
<th>features (genes)</th>
<th>patient #7</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESR1</td>
<td>876.4</td>
</tr>
<tr>
<td>ERBB2</td>
<td>100.1</td>
</tr>
<tr>
<td>BRCA1</td>
<td>798.8</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

classifier algorithm:

if (ESR1 > 100) AND (ERBB2 > 550) tumor = low-risk
else tumor = aggressive

predicted class

low-risk tumor
or
aggressive tumor
How is classification used with microarrays?

... mostly medical applications (?)

SNP profile of healthy tissue >> predict risk of disease

Expression profile of tissue, blood, etc. >> diagnose disease

Expression profile of tumor >> select optimal chemotherapy

Genomic profile of bacterial sample >> identification of species

... others?
Classifiers sound great

Where do I get one?
Two approaches to deriving classifiers

1. Simple “manual” methods (often univariate)
   – easy to understand
   – easy to derive
   – easy to evaluate

2. Machine learning methods (often multivariate)
   – complicated
   – better performance

Even though the exercises emphasize #2, I recommend using #1 if possible!
Hypothetical classification problem: gender

Input: Expression profile

Classifier

Output: Gender (male/female)
A very simple classifier

One gene!

If XIST > 7
  sex = female
else
  sex = male

Friday, January 11, 2013
Another simple classifier: “signature”

The average of 25 correlated genes

High = bad prognosis
Low = good prognosis
Recipe for a simple univariate classifier

1. Choose a set of genes
   a. Genes that you already know are relevant.
   or
   b. Genes that are differentially expressed between the two groups.
      \((t\) test, etc.)

2. Choose a simple rule for combining the gene measurements
   (mean, median, etc.)

3. Choose a decision threshold (cutoff).
Now, a complicated classifier

**A 21-gene recurrence score for breast cancer**

The recurrence score on a scale from 0 to 100 is derived from the reference-normalized expression measurements in four steps. First, expression for each gene is normalized relative to the expression of the five reference genes (ACTB [the gene encoding β-actin], GAPDH, GUS, RPLPO, and TFRC). Reference-normalized expression measurements range from 0 to 15, with a 1-unit increase reflecting approximately a doubling of RNA. Genes are grouped on the basis of function, correlated expression, or both. Second, the GRB7, ER, proliferation, and invasion group scores are calculated from individual gene-expression measurements, as follows: GRB7 group score = 0.9 × GRB7 + 0.1 × HER2 (if the result is less than 8, then the GRB7 group score is considered 8); ER group score = (0.8 × ER + 1.2 × PGR + BCL2 + SCUBE2) ÷ 4; proliferation group score = (Survivin + KI67 + MYBL2 + CCNB1 [the gene encoding cyclin B1] + STK15) ÷ 5 (if the result is less than 6.5, then the proliferation group score is considered 6.5); and invasion group score = (CTSL2 [the gene encoding cathepsin L2] + MMP11 [the gene encoding stromolysin 3]) ÷ 2. The unscaled recurrence score (RSU) is calculated with the use of coefficients that are predefined on the basis of regression analysis of gene expression and recurrence in the three training studies: RSU = +0.47 × GRB7 group score - 0.34 × ER group score + 1.04 × proliferation group score + 0.10 × invasion group score + 0.05 × CD68 - 0.08 × GSTM1 - 0.07 × BAG1. A plus sign indicates that increased expression is associated with an increased risk of recurrence, and a minus sign indicates that increased expression is associated with a decreased risk of recurrence. Fourth, the recurrence score (RS) is rescaled from the unscaled recurrence score, as follows: RS = 0 if RSU < 0; RS = 20 × (RSU × 6.7) if 0 ≤ RSU ≤ 100; and RS = 100 if RSU > 100.

...a complicated, nonlinear function of several variables!!

Machine learning

Let the computer do the work developing the classifier!

The goal: Given a set of *training data* (data of known class), develop a classifier that accurately predicts the class of novel data.

<table>
<thead>
<tr>
<th>gene</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>#7</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESR1</td>
<td>143.5</td>
<td>165.0</td>
<td>134.2</td>
<td>522.4</td>
<td>599.1</td>
<td>288.3</td>
<td>209.0</td>
</tr>
<tr>
<td>ERBB2</td>
<td>801.1</td>
<td>291.7</td>
<td>293.1</td>
<td>827.3</td>
<td>980.0</td>
<td>155.5</td>
<td>128.1</td>
</tr>
<tr>
<td>BRCA1</td>
<td>129.1</td>
<td>238.0</td>
<td>883.3</td>
<td>281.0</td>
<td>95.1</td>
<td>385.2</td>
<td>383.4</td>
</tr>
</tbody>
</table>

| aggressive | no | yes | yes | no | no | no | yes | ??? |

Training set
Classification as mapping

Suppose we want to develop a classifier based on the expression levels of two genes, G1 and G2:

Training data set:
- orange dots: aggressive tumor
- blue dots: benign tumor
- black dot: unknown

>>> How do we do this algorithmically?
Linear vs. nonlinear
Multiple dimensions...?
Recipe for classification with machine learning

1. Choose a classification method (simpler is probably better)

2. Feature selection / dimension reduction (fewer features is probably better)

3. Train the classifier

4. Assess the performance of the classifier
1. Choose a classification method

Linear:
- Linear discriminant analysis (LDA)
- Nearest centroid

Nonlinear:
- $k$ nearest neighbors (kNN)
- Artificial neural network (ANN)
- Support vector machine (SVM)

(many others also exist!)
Linear discriminant analysis (LDA)

Model the density of points as Gaussian.
Using model, find the optimal separating line (or hyperplane, in higher dimensions)

Assumptions:
• sampled from a normal distribution
• variance/covariance same in each class

R: lda (MASS package)
Nearest centroid

Calculate centroids for each class.
Class is predicted according to nearest centroid.

Similar to LDA.
Can be extended to multiple (n > 2) classes.
**k nearest neighbors (kNN)**

For a test case, find the *k* nearest samples in the training set, and let them vote.

- need to choose *k* (hyperparameter)
- *k* must be odd
- need to choose distance metric

No real “learning” involved - the training set defines the classifier.

R: `knn` *(class package)*
Artificial neural network (ANN)

Each “neuron” is simply a function (usually nonlinear).

The “network” to the left just represents a series of nested functions.

Can be difficult to understand how the classifier works.

R: \texttt{nnet} (\texttt{nnet} package)
Support vector machine (SVM)

Find a line / plane / hyperplane that maximizes the margin. Often uses the “kernel trick” to map data to higher dimensions.

- very flexible
- many parameters

Results can be difficult to interpret.

R: \texttt{svm} (\texttt{e1071} package)
Do you need a nonlinear classifier?
Recipe for classification with machine learning

1. Choose a classification method

2. Feature selection / dimension reduction

3. Train the classifier

4. Assess classifier performance
Why do dimension reduction?

Previous slides: 2 genes X 16 samples.

Your data: **22000** genes X 31 samples.

More genes = more parameters in classifier. **Potential for over-training!!** aka *The Curse of Dimensionality.*

The problem with overtraining...

Accuracy

Number of parameters

Training set

Test set (novel data)

Friday, January 11, 2013
Dimension reduction

Two approaches:

1. Data transformation
   - Principal component analysis (PCA); use top components
   - Find clusters of genes; use their average.

2. Feature selection (gene selection)
   - Significant genes: t-test / ANOVA
   - High variance genes
   - Hypothesis-driven
Recipe for classification with machine learning

1. Choose a classification method

2. Feature selection / dimension reduction

3. Train the classifier

4. Assess classifier performance
Train the classifier

If you know the exact classifier you want to use: **just do it.**

but...

Usually, we want to try several classifiers *or* several variations of a classifier. e.g. number of features, $k$ in kNN, etc. (hyperparameters)

The problem:

Testing several classifiers, and then choosing the *best one* leads to selection bias (= **overtraining**). This is bad.

Instead we can use **cross-validation** to choose a classifier.
Cross-validation

- Data set
  - split
  - Temporary training set
  - Temporary testing set
- Train Classifier
  - Apply classifier to temporary testing set
  - Assess performance

Do this several times!

Each time, select different sample(s) for the temporary testing set.

If you are using cross-validation as an optimization step, choose the classifier (or classifier hyperparameters) that results in best performance.
Cross-validation

10 specimens of known class

Split into training/test sets

Test

Training

Accuracy = 0.5

Repeat for other combinations

1.0

0.0

0.5

1.0

Average accuracy = 0.6
Cross-validation example

Given data with 100 samples

5-fold cross-validation:
- Training: 4/5 of data (80 samples)
- Testing: 1/5 of data (20 samples)
- 5 different models and performance results

Leave-one-out cross-validation (LOOCV)
- Training: 99/100 of data (99 samples)
- Testing: 1/100 of data (1 sample)
- 100 different models and performance results
Problems with cross-validation

1. You cannot use cross-validation to both optimize and evaluate your classifier.

2. The classifier obtained at each round of cross-validation is different, and is different from that obtained using all data.

3. Cross-validation will overestimate performance in the presence of experimental bias.

.... therefore an independent test set is *always* better!
Recipe for classification

1. Choose a classification method

2. Feature selection / dimension reduction

3. Train the classifier

4. Assess classifier performance
Assess performance of the classifier

How well does the classifier perform on novel samples?

Bad: Assess performance on the training set.

Alternative: Set aside a subset of your data as a test set.

Best: Also assess performance on an entirely independent data set (e.g. data produced by another lab).
Assessing classifier performance

### predict relapse of cancer

<table>
<thead>
<tr>
<th>Patient ID</th>
<th>predicted relapse</th>
<th>actual relapse</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>#001</td>
<td>yes</td>
<td>no</td>
<td>FP</td>
</tr>
<tr>
<td>#002</td>
<td>no</td>
<td>no</td>
<td>TN</td>
</tr>
<tr>
<td>#003</td>
<td>yes</td>
<td>yes</td>
<td>TP</td>
</tr>
<tr>
<td>#004</td>
<td>no</td>
<td>yes</td>
<td>FN</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

TP = True Positive  
TN = True Negative  
FP = False Positive  
FN = False Negative

### confusion matrix

<table>
<thead>
<tr>
<th></th>
<th>predict yes</th>
<th>predict no</th>
</tr>
</thead>
<tbody>
<tr>
<td>actual yes</td>
<td>131 TP</td>
<td>21 FN</td>
</tr>
<tr>
<td>actual no</td>
<td>7 FP</td>
<td>212 TN</td>
</tr>
</tbody>
</table>

Friday, January 11, 2013
Classifier performance: Accuracy

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}
\]

This is the fraction of predictions that are correct

range: [0 .. 1]

confusion matrix

<table>
<thead>
<tr>
<th></th>
<th>predict yes</th>
<th>predict no</th>
</tr>
</thead>
<tbody>
<tr>
<td>actual yes</td>
<td>131 TP</td>
<td>21 FN</td>
</tr>
<tr>
<td>actual no</td>
<td>7 FP</td>
<td>212 TN</td>
</tr>
</tbody>
</table>

Accuracy = 0.92
Classifier performance: sensitivity/specificity

\[
\text{Sensitivity} = \frac{TP}{TP + FN}
\]
the ability to detect positives

\[
\text{Specificity} = \frac{TN}{TN + FP}
\]
the ability to reject negatives

range: [0 .. 1]

confusion matrix (cancer)

<table>
<thead>
<tr>
<th></th>
<th>predict yes</th>
<th>predict no</th>
</tr>
</thead>
<tbody>
<tr>
<td>actual yes</td>
<td>131 TP</td>
<td>21 FN</td>
</tr>
<tr>
<td>actual no</td>
<td>7 FP</td>
<td>212 TN</td>
</tr>
</tbody>
</table>

Sensitivity = 0.86
Specificity = 0.97
Classifier performance: Matthews correlation coefficient

\[ MCC = \frac{(TP \cdot TN) - (FN \cdot FP)}{\sqrt{(TN + FN)(TN + FP)(TP + FN)(TP + FP)}} \]

A single performance measure that is less influenced by unbalanced test sets

range: [-1 .. 1]

confusion matrix (cancer)

<table>
<thead>
<tr>
<th></th>
<th>predict yes</th>
<th>predict no</th>
</tr>
</thead>
<tbody>
<tr>
<td>actual yes</td>
<td>131 TP</td>
<td>21 FN</td>
</tr>
<tr>
<td>actual no</td>
<td>7 FP</td>
<td>212 TN</td>
</tr>
</tbody>
</table>

\[ MCC = 0.84 \]
The problem with unbalanced test data

A new classifier: does patient have tuberculosis?

confusion matrix

<table>
<thead>
<tr>
<th></th>
<th>predict yes</th>
<th>predict no</th>
</tr>
</thead>
<tbody>
<tr>
<td>actual yes</td>
<td>5345 TP</td>
<td>21 FN</td>
</tr>
<tr>
<td>actual no</td>
<td>113 FP</td>
<td>18 TN</td>
</tr>
</tbody>
</table>

Accuracy = 0.98

For patients who do not really have tuberculosis: this classifier is usually wrong!

>>> Accuracy can be misleading, especially when the test cases are unbalanced
Evaluating unbalanced test data

The tuberculosis classifier

<table>
<thead>
<tr>
<th></th>
<th>predict yes</th>
<th>predict no</th>
</tr>
</thead>
<tbody>
<tr>
<td>actual yes</td>
<td>5345 TP</td>
<td>21 FN</td>
</tr>
<tr>
<td>actual no</td>
<td>113 FP</td>
<td>18 TN</td>
</tr>
</tbody>
</table>

Accuracy = 0.98
Sensitivity = 0.996
Specificity = 0.14
MCC = 0.24
Other performance metrics: Survival

**a.**

Gene low

Gene high

HR = 2.70 (1.00 - 7.00)

$P = 0.040$

**b.**

Gene low

Gene high

HR = 1.05 (0.59 - 1.86)

$P = 0.88$

<table>
<thead>
<tr>
<th>Time (years)</th>
<th>Number at risk</th>
<th>Number at risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>25</td>
<td>68</td>
</tr>
<tr>
<td>2</td>
<td>23</td>
<td>60</td>
</tr>
<tr>
<td>4</td>
<td>21</td>
<td>46</td>
</tr>
<tr>
<td>6</td>
<td>19</td>
<td>39</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>29</td>
</tr>
<tr>
<td>0</td>
<td>25</td>
<td>69</td>
</tr>
<tr>
<td>2</td>
<td>22</td>
<td>53</td>
</tr>
<tr>
<td>4</td>
<td>14</td>
<td>47</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>44</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>31</td>
</tr>
</tbody>
</table>
Other performance metrics: ROC curve

What if we don’t know where to set the threshold?

ROC = “Receiver operating characteristics”

AUC = “Area under the (ROC) curve”

AUC = 0.965

XIST

Sensitivity

1 - specificity

female

male
Summary: classification with machine learning

1. Choose a classification method
   – kNN, LDA, SVM, etc.

2. Feature selection / dimension reduction
   – PCA, possibly t-tests

3. Train the classifier
   – use cross-validation to optimize parameters

4. Assess classifier performance
   – use an independent test set if possible
   – determine sensitivity and specificity when appropriate
The data set in the exercise

Genome-wide expression profiling of human blood reveals biomarkers for Huntington's disease.


31 samples:
- 14 normal
- 5 presymptomatic
- 12 symptomatic

Affymetrix HG-U133A arrays

Huntington’s disease:
- neurological disorder
- polyglutamine expansion in the huntingtin gene

Why search for marker of disease progression (not diagnosis)?
- assess treatment efficacy
- surrogate endpoint in drug trials