Classification with microarray data

Aron Charles Eklund
Outline

Classification in general
  – overview
  – evaluation
  – algorithms
  – approach

Classification using microarray data
  – dimensionality

Example - childhood leukemia
What is classification?

Given an object with a set of features (input data), a classifier assigns the objects to a class.

<table>
<thead>
<tr>
<th>features</th>
<th>object</th>
<th>classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>Aron</td>
<td>democrat</td>
</tr>
<tr>
<td>sex</td>
<td>male</td>
<td>republican</td>
</tr>
<tr>
<td>income</td>
<td>$$$$</td>
<td></td>
</tr>
</tbody>
</table>

E.g.

if (age > 40) AND (income > $$)
    then vote = republican
else vote = democrat
The challenge

Given a *training set* (data of known class), design a classifier that *accurately* predicts the class of novel data.

<table>
<thead>
<tr>
<th>x01</th>
<th>x02</th>
<th>x03</th>
<th>x04</th>
<th>x05</th>
<th>x06</th>
<th>x07</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>43</td>
<td>65</td>
<td>34</td>
<td>22</td>
<td>44</td>
<td>29</td>
</tr>
<tr>
<td>sex</td>
<td>female</td>
<td>male</td>
<td>male</td>
<td>female</td>
<td>female</td>
<td>male</td>
</tr>
<tr>
<td>income</td>
<td>$$$$</td>
<td>$$$$</td>
<td>$$$$$</td>
<td>$</td>
<td>$$$$$</td>
<td>$</td>
</tr>
</tbody>
</table>
Classification algorithms

- Linear Discriminant Analysis (LDA)
- k Nearest Neighbors (kNN)
- Nearest Centroid

- Support Vector Machine (SVM)
- Artificial Neural Network (ANN)
Classification as mapping

A classifier based on the expression levels of two genes, G1 and G2:

For a patient with expression levels (G1, G2), what does classifier predict?

Training data set:
- Orange: patient voted for democrat
- Blue: patient voted for republican

>>> How do we do this algorithmically?
Linear Discriminant Analysis (LDA)

Find a line / plane / hyperplane

assumptions:
- normally distributed
- variance/covariance same in each class

appropriate only if data in linearly separable
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.
Nearest centroid

Calculate centroids for each class.

\[ \overline{x}_{ik} = \frac{\sum_{j \in C_k} x_{ij}}{n_k} \]

Similar to LDA.

Can be extended to multiple (n > 2) classes.
Artificial Neural Network (ANN)

Gene1  Gene2  Gene3  ...  

Input layer

Hidden layer

Output layer

Class A  Class B
## Comparison of methods

<table>
<thead>
<tr>
<th>Linear discriminant analysis</th>
<th>Neural networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nearest centroid</td>
<td>Support vector machines</td>
</tr>
<tr>
<td>KNN</td>
<td>Advanced methods</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Simple method</th>
<th>Advanced methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Based on distance calculation</td>
<td>Involve machine learning</td>
</tr>
<tr>
<td>Good for simple problems</td>
<td>Several adjustable parameters</td>
</tr>
<tr>
<td>Good for few training samples</td>
<td>Many training samples required (e.g., 50-100)</td>
</tr>
<tr>
<td>Distribution of data assumed</td>
<td>Flexible methods</td>
</tr>
</tbody>
</table>
Dimension reduction

Avoid overfitting!
Want: number of training points > number of features

• by transformation
  • PCA

• by feature selection (gene selection)
  • Significant genes: t-test / ANOVA
  • Selection of a limited number of genes
Cross-validation

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
Assessing classifier performance

predict relapse of cancer

<table>
<thead>
<tr>
<th></th>
<th>prediction</th>
<th>actual outcome</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Aron</td>
<td>yes</td>
<td>no</td>
<td>FP</td>
</tr>
<tr>
<td>Beth</td>
<td>no</td>
<td>no</td>
<td>TN</td>
</tr>
<tr>
<td>Chuck</td>
<td>yes</td>
<td>yes</td>
<td>TP</td>
</tr>
<tr>
<td>Dorthe</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>
Classifier performance: Accuracy

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

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>

\[ \text{Accuracy} = 0.92 \]
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
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: \([0 \ldots 1]\)

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>

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

<table>
<thead>
<tr>
<th>Predicted</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>

\[
MCC = 0.84
\]
### 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 (?)
- **Sensitivity** = 0.996
- **Specificity** = 0.14
- **MCC** = 0.24
“Honest” classifier assessment

1. Use separate data sets for **training** and for **testing**.

2. Do not “wear out” the testing data set.
Overview of Classification

Expression data

Subdivision of data for cross-validation into training sets and test sets

Feature selection (t-test)
Dimension reduction (PCA)

Training of classifier:
- using cross-validation
- choice of method
- choice of optimal parameters

Testing of classifier

Independent test set
Important Points

• Avoid over fitting

• Validate performance
  – Test on an independent test set
  – Use cross-validation

• Include feature selection in cross-validation

Why?
  – To avoid overestimation of performance!
  – To make a general classifier
Childhood Leukemia

- Cancer in the cells of the immune system
- Approx. 35 new cases in Denmark every year
- 50 years ago – all patients died
- Today – approx. 78% are cured
- Risk groups:
  - Standard, Intermediate, High, Very high, Extra high
- Treatment
  - Chemotherapy
  - Bone marrow transplantation
  - Radiation
# Prognostic Factors

<table>
<thead>
<tr>
<th></th>
<th>Good prognosis</th>
<th>Poor prognosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Immuno-phenotype</td>
<td>precursor B</td>
<td>T</td>
</tr>
<tr>
<td>Age</td>
<td>1-9</td>
<td>≥10</td>
</tr>
<tr>
<td>Leukocyte count</td>
<td>Low (&lt;50*10^9/L)</td>
<td>High (&gt;100*10^9/L)</td>
</tr>
<tr>
<td>Number of chromosomes</td>
<td>Hyperdiploidy (&gt;50)</td>
<td>Hypodiploidy (&lt;46)</td>
</tr>
<tr>
<td>Translocations</td>
<td>t(12;21)</td>
<td>t(9;22), t(1;19)</td>
</tr>
<tr>
<td>Treatment response</td>
<td>Good response: Low MRD</td>
<td>Poor response: High MRD</td>
</tr>
</tbody>
</table>
Risk Classification Today

Patient:
- Clinical data
- Immunophenotype
- Morphology
- Genetic measurements
- Microarray technology

Prognostic factors:
- Immunophenotype
- Age
- Leukocyte count
- Number of chromosomes
- Translocations
- Treatment response

Risk group:
- Standard
- Intermediate
- High
- Very high
- Extra High
Study of Childhood Leukemia

- Diagnostic bone marrow samples from leukemia patients
- Platform: Affymetrix Focus Array
  - 8793 human genes
- Immunophenotype
  - 18 patients with precursor B immunophenotype
  - 17 patients with T immunophenotype
- Outcome 5 years from diagnosis
  - 11 patients with relapse
  - 18 patients in complete remission
Study of Childhood Leukemia: Results

• Classification of immunophenotype (precursor B and T)
  – 100% accuracy
    • During the training
    • When testing on an independent test set
  – Simple classification methods applied
    • K-nearest neighbor
    • Nearest centroid

• Classification of outcome (relapse or remission)
  – 78% accuracy (CC = 0.59)
  – Simple and advanced classification methods applied