Themes of Bioinformatics

Bioinformatics =
  Data Mgmt +
  Knowledge Discovery +
  Sequence Analysis +
  Physical Modeling + ....

Knowledge Discovery =
  Statistics + Algorithms + Databases

Applications include diagnosis, prognosis, &
treatment optimization, often thru biomarker
discovery
What is Knowledge Discovery?

Jonathan’s blocks

Jessica’s blocks

Whose block is this?

Jonathan’s rules: Blue or Circle
Jessica’s rules: All the rest

Question: Can you explain how?
Key Steps of Knowledge Discovery

- Training data gathering

- Feature generation
  - k-grams, colour, texture, domain know-how, ...

- Feature selection
  - Entropy, $\chi^2$, CFS, t-test, domain know-how...

- Feature integration
  - SVM, ANN, PCL, CART, C4.5, kNN, ...

What is Accuracy?
What is Accuracy?

<table>
<thead>
<tr>
<th></th>
<th>predicted as positive</th>
<th>predicted as negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>positive</td>
<td>TP</td>
<td>FN</td>
</tr>
<tr>
<td>negative</td>
<td>FP</td>
<td>TN</td>
</tr>
</tbody>
</table>

Accuracy = \( \frac{\text{No. of correct predictions}}{\text{No. of predictions}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \)

Examples (Balanced Population)

<table>
<thead>
<tr>
<th>classifier</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>50%</td>
</tr>
<tr>
<td>B</td>
<td>50</td>
<td>25</td>
<td>25</td>
<td>0</td>
<td>75%</td>
</tr>
<tr>
<td>C</td>
<td>25</td>
<td>50</td>
<td>0</td>
<td>25</td>
<td>75%</td>
</tr>
<tr>
<td>D</td>
<td>37</td>
<td>37</td>
<td>13</td>
<td>13</td>
<td>74%</td>
</tr>
</tbody>
</table>

- Clearly, B, C, D are all better than A
- Is B better than C, D?
- Is C better than B, D?
- Is D better than B, C?

Accuracy may not tell the whole story
Examples (Unbalanced Population)

<table>
<thead>
<tr>
<th>classifier</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>25</td>
<td>75</td>
<td>75</td>
<td>25</td>
<td>50%</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>150</td>
<td>0</td>
<td>50</td>
<td>75%</td>
</tr>
<tr>
<td>C</td>
<td>50</td>
<td>0</td>
<td>150</td>
<td>0</td>
<td>25%</td>
</tr>
<tr>
<td>D</td>
<td>30</td>
<td>100</td>
<td>50</td>
<td>20</td>
<td>65%</td>
</tr>
</tbody>
</table>

- Clearly, D is better than A
- Is B better than A, C, D?

Exercise: What is B’s Prediction strategy?

What is Sensitivity (aka Recall)?

<table>
<thead>
<tr>
<th></th>
<th>predicted as positive</th>
<th>predicted as negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>positive</td>
<td>TP</td>
<td>FN</td>
</tr>
<tr>
<td>negative</td>
<td>FP</td>
<td>TN</td>
</tr>
</tbody>
</table>

\[
\text{Sensitivity} = \frac{\text{No. of correct positive predictions}}{\text{No. of positives}} = \frac{TP}{TP + FN}
\]

Sometimes sensitivity wrt negatives is termed specificity.
What is Precision?

<table>
<thead>
<tr>
<th></th>
<th>predicted as positive</th>
<th>predicted as negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>positive</td>
<td>TP</td>
<td>FN</td>
</tr>
<tr>
<td>negative</td>
<td>FP</td>
<td>TN</td>
</tr>
</tbody>
</table>

\[
\text{Precision} = \frac{\text{No. of correct positive predictions}}{\text{No. of positives predictions}} = \frac{\text{TP}}{\text{TP} + \text{FP}}
\]

Unbalanced Population Revisited

<table>
<thead>
<tr>
<th>classifier</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>Accuracy</th>
<th>Sensitivity</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>25</td>
<td>75</td>
<td>75</td>
<td>25</td>
<td>50%</td>
<td>50%</td>
<td>25%</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>150</td>
<td>0</td>
<td>50</td>
<td>75%</td>
<td>0%</td>
<td>ND</td>
</tr>
<tr>
<td>C</td>
<td>50</td>
<td>0</td>
<td>150</td>
<td>0</td>
<td>25%</td>
<td>100%</td>
<td>25%</td>
</tr>
<tr>
<td>D</td>
<td>30</td>
<td>100</td>
<td>50</td>
<td>20</td>
<td>65%</td>
<td>60%</td>
<td>38%</td>
</tr>
</tbody>
</table>

- What are the sensitivity and precision of B and C?
- Is B better than A, C, D?
Abstract Model of a Classifier

- Given a test sample $S$
- Compute scores $p(S)$, $n(S)$
- Predict $S$ as negative if $p(S) / n(S) < t$
- Predict $S$ as positive if $p(S) / n(S) \geq t$

$t$ is the decision threshold of the classifier
changing $t$ affects the recall and precision,
and hence accuracy, of the classifier

An Example

<table>
<thead>
<tr>
<th>$S$</th>
<th>$p(S)$</th>
<th>$n(S)$</th>
<th>Actual Class</th>
<th>Predicted Class $t = 3$</th>
<th>Predicted Class $t = 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.961252</td>
<td>0.038748</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>3</td>
<td>0.435302</td>
<td>0.564698</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>6</td>
<td>0.691596</td>
<td>0.308404</td>
<td>P</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>7</td>
<td>0.180805</td>
<td>0.819195</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>8</td>
<td>0.814909</td>
<td>0.185091</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>10</td>
<td>0.887220</td>
<td>0.112780</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>accuracy</th>
<th>recall</th>
<th>precision</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$t = 3$</td>
<td>5/6</td>
<td>3/4</td>
<td>4/4</td>
</tr>
<tr>
<td></td>
<td>$t = 2$</td>
<td>6/6</td>
<td>4/4</td>
<td>4/4</td>
</tr>
</tbody>
</table>

Recall that …
- Predict $S$ as negative if $p(S) / n(S) < t$
- Predict $S$ as positive if $p(S) / n(S) \geq t$
Comparing Prediction Performance

• **Accuracy is the obvious measure**
  – But it conveys the right intuition only when the positive and negative populations are roughly equal in size

• **Recall and precision together form a better measure**
  – But what do you do when A has better recall than B and B has better precision than A?

So let us look at some alternate measures ….

Adjusted Accuracy

• **Weigh by the importance of the classes**

  \[
  \text{Adjusted accuracy} = \alpha \times \text{Sensitivity} + \beta \times \text{Specificity}
  \]

  where \( \alpha + \beta = 1 \)

  typically, \( \alpha = \beta = 0.5 \)

<table>
<thead>
<tr>
<th>classifier</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>Accuracy</th>
<th>Adj Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>25</td>
<td>75</td>
<td>75</td>
<td>25</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>150</td>
<td>0</td>
<td>50</td>
<td>75%</td>
<td>50%</td>
</tr>
<tr>
<td>C</td>
<td>50</td>
<td>0</td>
<td>150</td>
<td>0</td>
<td>25%</td>
<td>50%</td>
</tr>
<tr>
<td>D</td>
<td>30</td>
<td>100</td>
<td>50</td>
<td>20</td>
<td>65%</td>
<td>63%</td>
</tr>
</tbody>
</table>

But people can’t always agree on values for \( \alpha, \beta \)
ROC Curves

- By changing $t$, we get a range of sensitivities and specificities of a classifier.
- A predicts better than B if A has better sensitivities than B at most specificities.
- Leads to ROC curve that plots sensitivity vs. $(1 - \text{specificity})$.
- Then the larger the area under the ROC curve, the better.

What is Cross Validation?
Construction of a Classifier

Training samples → Build Classifier → Classifier

Test instance → Apply Classifier → Prediction

Estimate Accuracy: Wrong Way

Training samples → Build Classifier → Classifier

Apply Classifier → Predictions

Estimate Accuracy → Accuracy

Exercise: Why is this way of estimating accuracy wrong?
K-Nearest Neighbour Classifier (k-NN)

• Assume S is well approximated by its neighbours
• Then, given a sample S, find the k observations $S_1 \ldots S_k$ in the known data that are “closest” to it, and average their responses

$$p(S) = \sum_{S_i \in N_k(S) \cap D^p} 1$$

$$n(S) = \sum_{S_i \in N_k(S) \cap D^n} 1$$

where $N_k(S)$ is the neighbourhood of S defined by the k nearest samples to it.

Assume distance between samples is Euclidean distance for now.

Illustration of kNN (k=8)
Estimate Accuracy: Wrong Way

Training samples → Build 1-NN → 1-NN

Apply 1-NN → Predictions

Estimate Accuracy → 100% Accuracy

Exercise:
Why does 1-NN has 100% accuracy under this scenario?

For sure k-NN (k = 1) has 100% accuracy in the “accuracy estimation” procedure above. But does this accuracy generalize to new test instances?

Estimate Accuracy: Right Way

Training samples → Build Classifier → Classifier

Testing samples → Apply Classifier → Predictions

Estimate Accuracy → Accuracy

Testing samples are NOT to be used during “Build Classifier”
Cross Validation

- Divide samples into k roughly equal parts
- Each part has similar proportion of samples from different classes
- Use each part to test other parts

Curse of Dimensionality
Curse of Dimensionality

- How much of each dimension is needed to cover a proportion $r$ of total sample space?

- Calculate by $e_p(r) = r^{1/p}$

- So, to cover 10% of a 15-D space, need to sample $(0.1)^{1/15} = 85\%$ of each dimension!

Exercise: Why $e_p(r) = r^{1/p}$?

Consequence of the Curse

- Suppose the number of samples given to us in the total sample space is fixed

- Let the dimension increase

- Then the distance of the $k$ nearest neighbours of any point increases

- Then the $k$ nearest neighbours are less and less useful for prediction, and can confuse the $k$-NN classifier (and other types of classifiers as well)
What is Feature Selection?

Tackling the Curse

• Given a sample space of p dimensions

• It is possible that some dimensions are irrelevant

• Need to find ways to separate those dimensions (aka features) that are relevant (aka signals) from those that are irrelevant (aka noise)
Signal Selection (Basic Idea)

- Choose a feature with low intra-class distance
- Choose a feature with high inter-class distance

Signal Selection (e.g., t-statistics)

The t-stat of a signal is defined as

\[ t = \frac{|\mu_1 - \mu_2|}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \]

where \( \sigma_i^2 \) is the variance of that signal in class \( i \), \( \mu_i \) is the mean of that signal in class \( i \), and \( n_i \) is the size of class \( i \).

Suggestion: a modification to t-stats when \( n_1 \) and \( n_2 \) are small.
Self-fulfilling Oracle

- Construct artificial dataset with 100 samples, each with 100,000 randomly generated features and randomly assigned class labels
- Select 20 features with the best t-statistics (or other methods)
- Evaluate accuracy by cross validation using only the 20 selected features
- The resultant estimated accuracy can be ~90%
- But the true accuracy should be 50%, as the data were derived randomly

What Went Wrong?

- The 20 features were selected from the whole dataset
- Information in the held-out testing samples has thus been “leaked” to the training process
- The correct way is to re-select the 20 features at each fold; better still, use a totally new set of samples for testing
Concluding Remarks

What have we learned?

• **Methodology of data mining**
  – Feature generation, feature selection, feature integration

• **Evaluation of classifiers**
  – Accuracy, sensitivity, precision
  – Cross validation

• **Curse of dimensionality**
  – Feature selection concept
  – Self-fulfilling oracle
References


- David Hand et al., *Principles of Data Mining*, MIT Press, 2001

- Jinyan Li et al., Data Mining Techniques for the Practical Bioinformatician, *The Practical Bioinformatician*, Chapter 3, pages 35—70, WSPC, 2004