For written notes on this lecture, please read chapter 3 of *The Practical Bioinformatician*,

CS2220: Introduction to Computational Biology
Unit 1a: Essence of Knowledge Discovery

Wong Limsoon
What is knowledge discovery?

Jonathan’s blocks
- Hexagon
- Circle
- Triangle
- Cylinder

Jessica’s blocks
- Triangle
- Square
- Cylinder
- Diamond

Whose block is this?

Jonathan’s rules: Blue or Circle
Jessica’s rules: All the rest
What is knowledge discovery?

Question: Can you explain how?
Key steps

• 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, ...

Some classifiers / machine learning methods
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?

High accuracy is meaningless if population is unbalanced
What is sensitivity (aka recall)?

Sensitivity = \frac{\text{No. of correct positive predictions}}{\text{No. of positives}}

= \frac{TP}{TP + FN}

Sometimes sensitivity wrt negatives is termed **specificity**

Exercise: Write down the formula for 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>

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

\[
= \frac{TP}{TP + FP}
\]
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></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>50</td>
<td>0</td>
<td>150</td>
<td>0</td>
<td>25%</td>
<td></td>
<td></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?

Exercise #1
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
## Example

<table>
<thead>
<tr>
<th>$S$</th>
<th>$P(S)$</th>
<th>$N(S)$</th>
<th>Actual Class</th>
<th>Predicted Class $\theta t = 3$</th>
<th>Predicted Class $\theta t = 2$</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>P</td>
</tr>
<tr>
<td>7</td>
<td>0.180885</td>
<td>0.819115</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></td>
<td>P</td>
</tr>
</tbody>
</table>

**Accuracy** 5/6 6/6
**Recall** 3/4 4/4
**Precision** 3/3 4/4

Recall that …
- Predict $S$ as negative if $p(S) / n(S) < t$
- Predict $S$ as positive if $p(S) / n(S) \geq t$
Precision-recall trade-off

- A predicts better than B if A has better recall and precision than B
- There is a trade-off between recall and precision

- In some apps, once you reach satisfactory precision, you optimize for recall
- In some apps, once you reach satisfactory recall, you optimize for precision
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?
F-measure (Used in info extraction)

- Take the harmonic mean of recall and precision

\[
F = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \quad \text{(wrt positives)}
\]

<table>
<thead>
<tr>
<th>classifier</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>Accuracy</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>25</td>
<td>75</td>
<td>75</td>
<td>25</td>
<td>50%</td>
<td>33%</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>150</td>
<td>0</td>
<td>50</td>
<td>75%</td>
<td>undefined</td>
</tr>
<tr>
<td>C</td>
<td>50</td>
<td>0</td>
<td>150</td>
<td>0</td>
<td>25%</td>
<td>40%</td>
</tr>
<tr>
<td>D</td>
<td>30</td>
<td>100</td>
<td>50</td>
<td>20</td>
<td>65%</td>
<td>46%</td>
</tr>
</tbody>
</table>

Does not accord with intuition:
C predicts everything as +ve, but still rated better than A
Adjust the accuracy

- Weigh by the importance of the classes

Adjusted accuracy = $\alpha \times \text{Sensitivity} + B \times \text{Specificity}$

where $\alpha + \beta = 1$

typically, $\alpha = \beta = 0.5$

classifier | TP | TN | FP | FN | Accuracy | Adj Accuracy
--- | --- | --- | --- | --- | --- | ---
A | 25 | 75 | 75 | 25 | 50% | 50%
B | 0 | 150 | 0 | 50 | 75% | 50%
C | 50 | 0 | 150 | 0 | 25% | 50%
D | 30 | 100 | 50 | 20 | 65% | 63%
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.
Food for thought

• You have a classifier. On a test set having 20% +ve and 80% -ve cases, the classifier’s recall and precision are both 80%

• Suppose you test it on a new test set having 80% +ve and 20% -ve cases. What do you expect its accuracy to be?

• You may assume that the +ve (resp. –ve) cases in both test sets are equally sufficiently representative of the +ve (resp. –ve) real-world population

• What lesson have you learned?

Exercise #2
What is cross validation?
Construction of a classifier

Training samples → Build Classifier → Classifier

Test instance → Apply Classifier → Prediction
Estimate accuracy: Wrong way

- Why is this way of estimating accuracy wrong?

Exercise #3
Recall ...

...the 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
K-nearest neighbour classifier (k-NN)

• Given a sample S, find the k observations $S_i$ in the known data that are “closest” to it, and take majority vote of their responses

• Assume S is well approximated by its neighbours

\[ p(S) = \sum_{S_i \in N_k(S) \cap D^P} 1 \quad n(S) = \sum_{S_i \in N_k(S) \cap D^N} 1 \]

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)

Neighborhood

5 of class
3 of class

= Image credit: Zaki
Estimate accuracy: Wrong way

For sure k-NN (k = 1) has 100% accuracy (Why?) in the “accuracy estimation” procedure above. 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”
How many training and testing samples?

- No fixed ratio between training and testing samples; but typically 2:1 ratio

- Proportion of instances of different classes in testing samples should be similar to proportion in the real world, and preferably also to proportion in the training samples

- What if there are insufficient samples to reserve 1/3 for testing?
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
- Total up accuracy
How many fold?

• If samples are divided into k parts, we call this k-fold cross validation

• Choose k so that
  – the k-fold cross validation accuracy does not change much from k-1 fold
  – each part within the k-fold cross validation has similar accuracy

• $k = 5$ or $10$ are popular choices for $k$
Food for thought

• What is the logical basis of cross validation?
• Hint: Central limit theorem

• What / whose accuracy does it really estimate?
Curse of dimensionality
Recall kNN …

Neighborhood

- 5 of class
- 3 of class

Image credit: Zaki
Curse of dimensionality

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

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

- So, to cover 10% of a 15-D space, need 85% of each dimension!

Exercise #5
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
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-stats of a signal is defined as

\[ t = \frac{|\mu_1 - \mu_2|}{\sqrt{\frac{\sigma^2_1}{n_1} + \frac{\sigma^2_2}{n_2}}} \]

where \( \sigma^2_i \) 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 \( i \).
Food for thought

• How is the t-statistic typically used?

• What are the assumptions required for this way of using the t-statistic?
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 the 20 selected features

• The resulting 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 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
While **dimensionality reduction** is an important tool in machine learning/data mining, we must always be aware that it can distort the data in misleading ways.

Above is a two dimensional projection of an intrinsically three dimensional world....
A cloud of points in 3D

Can be projected into 2D

**XY** or **XZ** or **YZ**

In 2D **XZ** we see a triangle

In 2D **YZ** we see a square

In 2D **XY** we see a circle

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
Any questions?
Acknowledgements

- The first two slides were shown to WLS 20+ years ago by Tan Ah Hwee

- The three slides on the dangers of dimensionality reduction were created by Eamonn Keogh
References

• John A. Swets, Measuring the accuracy of diagnostic systems, Science 240:1285--1293, June 1988

• Trevor Hastie et al., The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, 2001. Chapters 1, 7

• Lance D. Miller et al., Optimal gene expression analysis by microarrays, Cancer Cell 2:353--361, 2002

• 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