Evaluating machine learning outcomes

- How should we define a good outcome for a machine learning algorithm?
- How do we know when an algorithm is doing well in a particular application?
- How can we compare different classifiers?
- How can we compare different learning algorithms?

What to Measure?

- More concrete questions:
  - What quantity should we measure?
  - How can we estimate it?
  - Can we get quantitative guarantees for estimates and comparisons?

Confusion Matrix

- \[ \frac{TP + TN}{TP + TN + FP + FN} \]
- In many applications one class (typically the +) has very low frequency
- Acc no longer the most appropriate criterion

Confusion Matrix

- \[ \frac{TP + TN}{TP + TN + FP + FN} \]
- In many applications one class (typically the +) has very low frequency
- Running algorithm to optimize Acc often yields bad results. Why?
IR Terminology

- **Context:** search for items with \texttt{QueryTerm}
- De-emphasize role of Neg examples
- Aims to measure quality of response
  
  \[
  \text{Precision} = \frac{TP}{TP + FP} \quad \text{Recall} = \frac{TP}{TP + FN}
  \]

- Tradeoff given by Precision/Recall curve
- And by \( F = \frac{2 \cdot R \cdot P}{R + P} \)

---

Medical Community Terminology

- **Context:** test to identify \texttt{Med Condition}
  
  \[
  \text{Sensitivity} = \text{Recall} = \frac{TP}{TP + FN} \quad \text{Specificity} = \frac{TN}{TN + FP}
  \]

- These measure "accuracy" measure within each class

---

Signal Detection Terminology

- **Context:** identify "signal"
  
  \[
  \text{TPrate} = \text{Recall} = \frac{TP}{TP + FN} \quad \text{FPrate} = 1 - \text{Specificity} = \frac{FP}{TN + FP}
  \]

- The ROC curve (receiver operator characteristic) plots TPrate vs FPrate

---

ROC Curves

- The AROC (area under ROC curve) has a nice interpretation as the the probability of correctly ranking a random Pos example above a random Neg example

---

From Ranking to Measures

- Many learning algorithms (e.g., Perceptron) provide a numerical output that can be used to rank examples in addition to the prediction of +/- label

- This can be used to produce a ROC curve for the algorithm by changing its threshold specifying the transition from Neg to Pos.

---

How to Measure?

- **Validation Set Method:**
  - keep aside a portion of the example set
  - Train model on remaining data
  - Measure performance on validation set

- Wastes data ...

- Unbiased estimate of quantity

- Variance in estimate due to choice of validation set. Can we fix this?
### How to Measure?

- **Validation Set Method:**
  - keep aside a portion of the example set
  - Train model on remaining data
  - Measure performance on validation set
  - Can reduce variance by repeating k times and averaging
  - But this introduces bias in estimates because the train/test data in different runs are highly correlated

### Leave One Out Method

- Setting k=N (number of examples) we get the leave one out method
- Often effective but
- High variance in individual estimates
- Expensive if we need to train N times (for some algorithms e.g., kNN can get away without this)

### Using Cross Validation

- Cross validation can be used to estimate how well algorithm does, and to compare algorithms - we develop this topic next
- But also ...

### Quantitative Comparisons

Review notions from Normal distributions

<table>
<thead>
<tr>
<th></th>
<th>50%</th>
<th>68%</th>
<th>80%</th>
<th>90%</th>
<th>95%</th>
<th>98%</th>
<th>99%</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu \pm \sigma$</td>
<td>0.67</td>
<td>1.00</td>
<td>1.28</td>
<td>1.64</td>
<td>1.96</td>
<td>2.33</td>
<td>2.58</td>
</tr>
</tbody>
</table>
### Quantitative Comparisons

- From probability to estimates
  - N% of area (probability) lies in $\mu \pm z_N \sigma$
- Probability Statement
  - [with probability N%] $x \in \mu \pm z_N \sigma$
- Confidence interval
  - [with confidence N%] $\mu \in x \pm z_N \sigma$

### Confidence Interval

- [with confidence N%] $\mu \in x \pm z_N \sigma$
- Assume $x \sim N(\mu, \sigma^2)$
- And we sample to observe value of $x$
  - [with confidence 95%] $\mu \in x \pm 1.96 \sigma$

### Evaluating one classifier

- We measure the performance on test set of $n$ examples
- If the error rate is $p$ then the number of mistakes on the test set is distributed as $\text{Binomial}(n, p)$
- Our estimate is $\hat{p} = \frac{\# \text{ mistakes}}{n}$
- $\hat{p}$ distributed approx as $\hat{p} \sim N(p, \frac{p(1-p)}{n})$
- Therefore, we can apply
  - [with confidence N%] $\mu \in x \pm z_N \sigma$
  - To get
    - [with confidence N%] $p \in \hat{p} \pm z_N \sqrt{\frac{p(1-p)}{n}}$
Evaluating Learning Algorithm

• Cross validation error estimate \( \hat{e} = \frac{1}{k} \sum e_i \)
• Define \( e \) as the expected error rate when running \( \text{alg} \) (on data of this size)
• Then we have:
  \[ e_i \sim N(e, \sigma_e^2) \]
  \[ \hat{e} \sim N(e, \frac{\sigma_e^2}{k}) \]

• And we can use
  \[
  \text{with confidence } N\% \quad e \in \hat{e} \pm z_N \sqrt{\frac{\sigma_e^2}{k}}
  \]

Evaluating Learning Algorithm

• But we do not know \( \sigma_e \)
• Solution 1: (don’t use this) plug in the estimate \( \hat{s} \) instead of \( \sigma_e \)
  \[
  s = \sqrt{\frac{1}{k-1} \sum (e_i - \hat{e})^2}
  \]
• Solution 2: a more accurate interval through \( T \) random variable

Evaluating Learning Algorithm

• It turns out [we skip definitions and details] that
  \[
  \frac{\hat{e} - e}{s / \sqrt{k}} \sim T_{k-1}
  \]
• And that this implies
  \[
  \text{with confidence } N\% \quad e \in \hat{e} \pm t_{N,k-1} \frac{s}{\sqrt{k}}
  \]
• And more concretely
  \[
  \text{with confidence } N\% \quad e \in \hat{e} \pm t_{N,k-1} \sqrt{\frac{\sum (e_i - \hat{e})^2}{(k-1)k}}
  \]

Comparing Two Algorithms

• We can calculate an interval for the error of each and say that one is better if the intervals do not overlap
• But we can do much better when variance \( \sigma_e \) is large
  • Alg1  Alg2  Diff
  • 60    68    8
  • 70    77    7
  • 80    88    8

Comparing Two Algorithms

• We can calculate an interval for the error of each and say that one is better if the intervals do not overlap
• But we can do much better when variance \( \sigma_e \) is large
• We can apply the same methodology and interval to the random variables representing the differences
• Apply to runs on the same folds reduces variance

<table>
<thead>
<tr>
<th>Alg1</th>
<th>Alg2</th>
<th>Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>68</td>
<td>8</td>
</tr>
<tr>
<td>70</td>
<td>77</td>
<td>7</td>
</tr>
<tr>
<td>80</td>
<td>88</td>
<td>8</td>
</tr>
</tbody>
</table>
Running many experiments and tests

- We develop 100 new algorithms
- Calculate interval with $N=95\%$ confidence for diff over baseline alg
- What is the probability of wrongly claiming a significant difference in at least one of these?

\[
1 - 0.95^{100} = 1 - 0.059 = 0.9941
\]

Running many experiments and tests

- We develop 100 new algorithms
- Calculate interval with $N=95\%$ confidence for diff over baseline alg
- To get 0.95 confidence for all the tests together we need each individual run with $N=99.95\%$

\[
1 - 0.9995^{100} = 1 - 0.9512 = 0.0488
\]