Data Mining II
Model Validation

Heiko Paulheim
Why Model Validation?

• We have seen so far
  – Various metrics (e.g., accuracy, F-measure, RMSE, …)
  – Evaluation protocol setups
    • Split Validation
    • Cross Validation
    • Special protocols for time series
    • …

• Today
  – A closer look at evaluation protocols
  – Asking for significance
  – Utilizing model explanations
Some Observations

- Data Mining Competitions often have a hidden test set
  - e.g., Data Mining Cup
  - e.g., many tasks on Kaggle
- Ranking on public test set and ranking on hidden test set may differ
- Example on one Kaggle competition:

https://www.kaggle.com/c/restaurant-revenue-prediction/discussion/14026
Some Observations: DMC 2018

- We had eight teams in Mannheim
- We submitted the results of the best and the third best(!) local team
  - The third best local team(!!!) got among the top 10
    - and eventually scored 2nd worldwide
- Meanwhile, the best local team did not get among the top 10
What is Happening Here?

- We have come across this problem quite a few times
- It’s called *overfitting*
  - Problem: we don’t know the error on the (hidden) test set

![Graph showing overfitting](https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/)

but according to the test set, we should have used that one

according to the training dataset, this model is the best one
Overfitting Revisited

- Typical DMC Setup:
  
  ![Diagram showing training and test data]

  - we often simulate test data by split or cross validation

- Possible overfitting scenarios:
  - our test partition may have certain characteristics
  - the “official” test data has different characteristics than the training data
Overfitting Revisited

- Typical Kaggle Setup:

  ![Diagram showing training data and test data]

  - Training Data
  - Test Data

- Possible overfitting scenarios:
  - solutions yielding good rankings on public leaderboard are preferred
  - models overfit to the public part of the test data

  undisclosed part of the test data used for private leaderboard
Overfitting Revisited

- Some flavors of overfitting are more subtle than others
- **Obvious overfitting:**
  - use test partition for training
- **Less obvious overfitting:**
  - tune parameters against test partition
  - select “best” approach based on test partition
- **Even less obvious overfitting**
  - use test partition in feature construction, for features such as
    - avg. sales of product per day
    - avg. orders by customer
    - computing trends
Overfitting Revisited

• Typical real world scenario:

Data from the past  |  The future (no data)

• Possible overfitting scenarios:
  – Similar to the DMC/Kaggle case, but worse
  – We do not even know the data on which we want to predict

we often simulate test data by split or cross validation
What Unlabeled Test Data can Tell Us

• If we have test data without labels, we can still look at predictions
  – do they look somehow reasonable?

• Task of DMC 2018: predict date of the month in which a product is sold out
  – Solutions for three best (local) solutions:
The Overtuning Problem

• In academia
  – many fields have their established benchmarks
  – achieving outstanding scores on those is required for publication
  – interesting novel ideas may score suboptimally
    • hence, they are not published
  – intensive tuning is required for publication
    • hence, available compute power often beats good ideas

• That “leaderboardism” has been criticized recently
The Overtuning Problem

• In real world projects
  – models overfit to past data
  – performance on unseen data is often overestimated
    • i.e., customers are disappointed
  – changing characteristics in data may be problematic
    • drift: e.g., predicting battery lifecycles
    • events not in training data: e.g., predicting sales for next month
  – cold start problem
    • some instances in the test set may be unknown before
    • e.g., predicting product sales for new products
Validating and Comparing Models

• When is a model good?
  – i.e., is it better than random?

• When is a model really better than another one?
  – i.e., is the performance difference by chance or by design?

Some of the following contents are taken from William W. Cohen’s Machine Learning Classes

http://www.cs.cmu.edu/~wcohen/
Confidence Intervals for Models

• Scenario:
  – you have learned a model M1 with an error rate of 0.30
  – the old model M0 had an error rate of 0.35
    (both evaluated on the same test set T)
• Do you think the new model is better?

• What might be suitable indicators?
  – size of the test set
  – model complexity
  – model variance
Size of the Test Set

• Scenario:
  – you have learned a model M1 with an error rate of 0.30
  – the old model M0 had an error rate of 0.35
    (both evaluated on the same test set S)

• Variant A: $|S| = 40$
  – a single error contributes 0.025 to the error rate
  – i.e., M1 got two more example right than M0

• Variant B: $|S| = 2,000$
  – a single error contributes 0.0005 to the error rate
  – i.e., M1 got 100 more examples right than M0
Size of the Test Set

• Scenario:
  – you have learned a model M1 with an error rate of 0.30
  – the old model M0 had an error rate of 0.35
    (both evaluated on the same test set S)

• Intuitively:
  – M1 is better if the error is observed on a larger test set S
  – The smaller the difference in the error, the larger |S| should be

• Can we formalize our intuitions?
What is an Error?

- Ultimately, we want to minimize the error on unseen data (D)
  - but we cannot measure it directly
- As a proxy, we use a sample S
  - in the best case: error\(_S\) = error\(_D\) ↔ |error\(_S\) − error\(_D\)| = 0
  - or, more precisely: \(E[|error\(_S\) − error\(_D\)|] = 0\) for each S

- In many cases, our models are overly optimistic
  - i.e., error\(_D\) > error\(_S\)
What is an Error?

• In many cases, our models are overly optimistic
  – i.e., $\text{error}_D > \text{error}_S$

• Most often, the model has overfit to $S$

• Possible reasons:
  – $S$ is a subset of training data (drastic)
  – $S$ has been used in feature engineering and/or parameter tuning
  – we have trained and tuned three models only on $T$, and pick the one which is best on $S$
What is an Error?

• Ultimately, we want to minimize the error on unseen data (D)
  – but we cannot measure it directly

• As a proxy, we use a sample S
  – unbiased model: \( E[|\text{error}_D - \text{error}_S|] = 0 \) for each S

• Even for an unbiased model, there is usually some variance given S
  – i.e. \( E[(\text{error}_S - E[\text{error}_S])^2] > 0 \)
  – intuitively: we measure (slightly) different errors on different S
Back to our Example

• Scenario:
  – you have learned a model M1 with an error rate of 0.30
  – the old model M0 had an error rate of 0.35
    (both evaluated on the same test set T)

• Old question:
  – is M1 better than M0?

• New question:
  – how likely is it the error of M1 is lower just by chance?
    • either: due to bias in M1, or due to variance
Back to our Example

• New question:
  – how likely is it the error of M1 is lower *just by chance*?
    • either: due to bias in M1, or due to variance

• Consider this a random process:
  – M1 makes an error on example x
  – Let us assume it actually has an error rate of 0.3
    • i.e., M1 follows a binomial with its maximum at 0.3

• Test:
  – what is the probability of actually observing 0.3 or 0.35 as error rates?
Binomial Distribution for M1

- We can easily construct those binomial distributions given \( n \) and \( p \)

\[
P(r) = \frac{n!}{r!(n-r)!} \cdot \text{error}_D(h)^r \cdot (1 - \text{error}_D(h))^{n-r}
\]

probability of observing an error of 0.3 (12/40): 0.137

probability of observing an error of 0.35 (14/40): 0.104
From the Binomial to Confidence Intervals

- New question:
  - what values are we likely to observe? (e.g., with a probability of 95%)
  - i.e., we look at the symmetric interval around the mean that covers 95%

\[
P(r) = \frac{n!}{r!(n-r)!} \cdot error_D(h)^r \cdot (1 - error_D(h))^{n-r}
\]
From the Binomial to Confidence Intervals

• With a probability of 95%, we observe 7 to 17 errors
  – corresponds to $[0.175 ; 0.425]$ as a confidence interval

• All observations *in* that interval are considered likely
  – i.e., an observed error rate of 0.35
    might also correspond to an actual error rate of 0.3

• Back to our example
  – on a test sample of $|S|=40$, we cannot say whether M1 or M0 is better
Simplified Calculation (z Test)

- The central limit theorem states that
  - a binomial distribution can be approximated by a Gaussian normal distribution
    - with $\mu = np$, $\sigma = \sqrt{\frac{p(1-p)}{n}}$
    - for sufficiently large $n$
      - rule of thumb: sufficiently large equals $n>30$
Simplified Calculation (z Test)

- The central limit theorem states that
  - a binomial distribution can be approximated by a Gaussian normal distribution
  - Gaussian distributions are simple to compute

80% of area (probability) lies in $\mu \pm 1.28\sigma$

N% of area (probability) lies in $\mu \pm z_N\sigma$

<table>
<thead>
<tr>
<th>N%</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>$z_N$</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>
Simplified Confidence Intervals

- Given that we have $|S| = n$, and an observed error $S$
  - With $p\%$ probability, error $D$ is in $[error_S - y, error_S + y]$
  - With $y = z_{N} \sqrt{\frac{error_S(1-error_S)}{n}}$

<table>
<thead>
<tr>
<th>$N%:$</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>$z_{N}:$</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>

- Given our example
  - $error_S = 0.30$, $n=40$
  - $\rightarrow$ with 95% probability, error $D$ is in $[0.158, 0.442]$
Working with Confidence Intervals

• Given that we have $|S|=n$, and an observed error $S$
  
  - With $p\%$ probability, error $D$ is in $[error_S - y, error_S + y]$
  
  - With $y=\frac{z_{p/2}\sqrt{error_S(1-error_S)}}{n}$

  Observation: the interval shrinks with growing $n$

• Recap: we had two scenarios, $|S| = 40$ and $|S| = 2000$
  
  - Interval for $n=40$: error $D$ is in $[0.158, 0.442]$
  
  - Interval for $n=2000$: error $D$ is in $[0.280, 0.320]$

• So, for $|S|=2000$, the probability that error $D$ is lower than 0.35 is >95%
Working with Confidence Intervals

• Comparing M0 and M1

|\(|S|=40\) |\(|S|=2000\) |
|---|---|
| ![Graph for |S|=40](image) | ![Graph for |S|=2000](image) |

• For \(|S|=2000\), the confidence intervals do not overlap
  – i.e., with 95% probability, M1 is better than M0
  – but we cannot make such a statement for \(|S|=40\)
Occam's Razor Revisited

- Named after William of Ockham (1287-1347)
- A fundamental principle of science
  - if you have two theories
  - that explain a phenomenon equally well
  - choose the simpler one

- Example:
  - phenomenon: the street is wet
  - theory 1: *it has rained*
  - theory 2: *a beer truck has had an accident, and beer has spilled. The truck has been towed, and magpies picked the glass pieces, so only the beer remains*
Occam's Razor Revisited

- Let’s rephrase:
  - if you have two models
  - where none is *significantly* better than the other
  - choose the simpler one

- Indicators for simplicity:
  - number of features used
  - number of variables used, e.g.,
    - hidden neurons in an ANN
    - no. of trees in a Random Forest
    - ...

Model Variance

• What happens if you repeat an experiment...
  – ...on a different test set?
  – ...on a different training set?
  – ...with a different random seed?

• Some methods may have higher *variance* than others
  – if your result was good, was just luck?
  – what is your actual estimate for the future?

• Typically, we need more than one experiment!
Model Variance

• Scenario:
  – you have learned a model M1 with an error rate of 0.30
  – the old model M0 had an error rate of 0.35
    (this time: in 10-fold cross validation)

• Variant A:
  – M0:

  \[
  \begin{array}{cccccccccc}
  F_1 & F_2 & F_3 & F_4 & F_5 & F_6 & F_7 & F_8 & F_9 & F_{10} & \phi \\
  0.37 & 0.28 & 0.38 & 0.40 & 0.27 & 0.42 & 0.26 & 0.39 & 0.41 & 0.29 & 0.35
  \end{array}
  \]

  – M1_A:

  \[
  \begin{array}{cccccccccc}
  F_1 & F_2 & F_3 & F_4 & F_5 & F_6 & F_7 & F_8 & F_9 & F_{10} & \phi \\
  0.28 & 0.30 & 0.31 & 0.32 & 0.25 & 0.32 & 0.27 & 0.32 & 0.33 & 0.30 & 0.30
  \end{array}
  \]
Model Variance

• Scenario:
  – you have learned a model M1 with an error rate of 0.30
  – the old model M0 had an error rate of 0.35
    (this time: in 10-fold cross validation)

• Variant B:
  – M0:
    | F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8 | F9 | F10 | Ø  |
    |---|---|---|---|---|---|---|---|---|----|---|
    | 0.37| 0.28| 0.38| 0.40| 0.27| 0.42| 0.26| 0.39| 0.41| 0.29| 0.35|

  – M1_B:
    | F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8 | F9 | F10 | Ø  |
    |---|---|---|---|---|---|---|---|---|----|---|
    | 0.17| 0.29| 0.18| 0.53| 0.28| 0.49| 0.27| 0.29| 0.19| 0.31| 0.30|

lucky shots

total fails
Model Variance

• M0:

\[
\begin{array}{cccccccccc}
F1 & F2 & F3 & F4 & F5 & F6 & F7 & F8 & F9 & F10 & \varnothing \\
0.37 & 0.28 & 0.38 & 0.40 & 0.27 & 0.42 & 0.26 & 0.39 & 0.41 & 0.29 & 0.35 \\
\end{array}
\]

• M1_A:

\[
\begin{array}{cccccccccc}
F1 & F2 & F3 & F4 & F5 & F6 & F7 & F8 & F9 & F10 & \varnothing \\
0.28 & 0.30 & 0.31 & 0.32 & 0.25 & 0.32 & 0.27 & 0.32 & 0.33 & 0.30 & 0.30 \\
\end{array}
\]

• M1_B:

\[
\begin{array}{cccccccccc}
F1 & F2 & F3 & F4 & F5 & F6 & F7 & F8 & F9 & F10 & \varnothing \\
0.17 & 0.29 & 0.18 & 0.53 & 0.28 & 0.49 & 0.27 & 0.29 & 0.19 & 0.31 & 0.30 \\
\end{array}
\]

• Some observations:
  – Standard deviations (M0: 0.06, M1_A: 0.03, M1_B: 0.12)
  – Pairwise competition:
    • M1_A outperforms M0 in 7/10 cases
    • but: M0 also outperforms M1_B in 6/10 cases!
  – Worst case of M1_A is below that of M0, but worst case of M1_B is above
Model Variance

• Why is model variance important?
  – recap: confidence intervals
  – risk vs. gain (use case!)
  – often, training data differs
    • even if you use cross or split validation during development
    • you might still train a model on the entire training data later
General Comparison of Methods

- Practice: finding a good method for a given problem
- Research: finding a good method for a *class of problems*

https://xkcd.com/664/
General Comparison of Methods

• Practice: finding a good method for a given problem
• Research: finding a good method for a class of problems

• Typical research paper:
  – Method M is better than state of the art S on a problem class P
  – Evaluation: show results of M on a subset of P
  – Claim that M is significantly better than S

let’s look closer
General Comparison of Methods

- De facto gold standard paper: Demšar, 2006
  - almost 10k citations on Google scholar
  - one of the most cited papers in JMLR in general
Example

- New Method M vs. State of the Art Method S
  - Tested on 12 different problems
  - Depicted: error rate
- Observations:
  - error rate alone might not be telling
  - problems are not directly comparable
Example

• Observation:
  – 9 times: M outperforms S
  – 2 times: S outperforms M
  – 1 tie

• Just looking at those outcomes
  – Null hypothesis: M and S are equally good
    • i.e., probability of M outperforming S is 0.5
  – What is the likelihood of M outperforming S in 9 or more out of 11 cases?
    • analogy: what is the likelihood of 9 or more heads in 11 coin tosses?

→ known as sign test

<table>
<thead>
<tr>
<th>Problem</th>
<th>M</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.09</td>
<td>0.11</td>
</tr>
<tr>
<td>2</td>
<td>0.71</td>
<td>0.72</td>
</tr>
<tr>
<td>3</td>
<td>0.77</td>
<td>0.69</td>
</tr>
<tr>
<td>4</td>
<td>0.21</td>
<td>0.44</td>
</tr>
<tr>
<td>5</td>
<td>0.37</td>
<td>0.37</td>
</tr>
<tr>
<td>6</td>
<td>0.85</td>
<td>0.92</td>
</tr>
<tr>
<td>7</td>
<td>0.62</td>
<td>0.65</td>
</tr>
<tr>
<td>8</td>
<td>0.58</td>
<td>0.55</td>
</tr>
<tr>
<td>9</td>
<td>0.79</td>
<td>0.89</td>
</tr>
<tr>
<td>10</td>
<td>0.12</td>
<td>0.16</td>
</tr>
<tr>
<td>11</td>
<td>0.09</td>
<td>0.15</td>
</tr>
<tr>
<td>12</td>
<td>0.19</td>
<td>0.24</td>
</tr>
<tr>
<td>Avg.</td>
<td>0.45</td>
<td>0.49</td>
</tr>
</tbody>
</table>
Example

- We’ve already seen something similar
  - what is the likelihood of that outcome (9/11 wins for M) by chance?
  - let’s look at confidence intervals

M wins: \( \frac{9}{11} \pm 1.96 \sqrt{\frac{\frac{9}{11} \times (1 - \frac{9}{11})}{11}} \rightarrow [0.59, 1.05] \)

S wins: \( \frac{2}{11} \pm 1.96 \sqrt{\frac{\frac{2}{11} \times (1 - \frac{2}{11})}{11}} \rightarrow [-0.05, 0.41] \)

- Looks safe, but... \( n < 30! \)
Example

• Observation:
  – 9 times: M outperforms S
  – 2 times: S outperforms M
  – 1 tie

• Just looking at those outcomes
  – Null hypothesis: M and S are equally good
    • i.e., probability of M outperforming S is 0.5
  – What is the likelihood of M outperforming S in 9 or more out of 11 cases?
    • analogy: what is the likelihood of 9 or more heads in 11 coin tosses?
  – Here: 0.03
    → i.e., with a probability >0.95, this is not an outcome by chance
Sign Test

- Observation:
  - 9 times: M outperforms S
  - 2 times: S outperforms M
  - 1 tie

- Sign test looks at those outcomes as binary experiments
  - null hypothesis: M is not better than S, i.e., M outperforming S is as likely as M not outperforming S

<table>
<thead>
<tr>
<th>Problem</th>
<th>M</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.09</td>
<td>0.11</td>
</tr>
<tr>
<td>2</td>
<td>0.71</td>
<td>0.72</td>
</tr>
<tr>
<td>3</td>
<td>0.77</td>
<td>0.69</td>
</tr>
<tr>
<td>4</td>
<td>0.21</td>
<td>0.44</td>
</tr>
<tr>
<td>5</td>
<td>0.37</td>
<td>0.37</td>
</tr>
<tr>
<td>6</td>
<td>0.85</td>
<td>0.92</td>
</tr>
<tr>
<td>7</td>
<td>0.62</td>
<td>0.65</td>
</tr>
<tr>
<td>8</td>
<td>0.58</td>
<td>0.55</td>
</tr>
<tr>
<td>9</td>
<td>0.79</td>
<td>0.89</td>
</tr>
<tr>
<td>10</td>
<td>0.12</td>
<td>0.16</td>
</tr>
<tr>
<td>11</td>
<td>0.09</td>
<td>0.15</td>
</tr>
<tr>
<td>12</td>
<td>0.19</td>
<td>0.24</td>
</tr>
<tr>
<td>Avg.</td>
<td>0.45</td>
<td>0.49</td>
</tr>
</tbody>
</table>

Table 3: Critical values for the two-tailed sign test at $\alpha = 0.05$ and $\alpha = 0.10$. A classifier is significantly better than another if it performs better on at least $w_{\alpha}$ data sets.
Sign Test – Variants

• Some variations:
  – We used N = wins + losses (standard sign test)
    some use: N= wins + losses + ties

• With that variant, we would not conclude significance at p<0.05

<table>
<thead>
<tr>
<th>Problem</th>
<th>M</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.09</td>
<td>0.11</td>
</tr>
<tr>
<td>2</td>
<td>0.71</td>
<td>0.72</td>
</tr>
<tr>
<td>3</td>
<td>0.77</td>
<td>0.69</td>
</tr>
<tr>
<td>4</td>
<td>0.21</td>
<td>0.44</td>
</tr>
<tr>
<td>5</td>
<td>0.37</td>
<td>0.37</td>
</tr>
<tr>
<td>6</td>
<td>0.85</td>
<td>0.92</td>
</tr>
<tr>
<td>7</td>
<td>0.62</td>
<td>0.65</td>
</tr>
<tr>
<td>8</td>
<td>0.58</td>
<td>0.55</td>
</tr>
<tr>
<td>9</td>
<td>0.79</td>
<td>0.89</td>
</tr>
<tr>
<td>10</td>
<td>0.12</td>
<td>0.16</td>
</tr>
<tr>
<td>11</td>
<td>0.09</td>
<td>0.15</td>
</tr>
<tr>
<td>12</td>
<td>0.19</td>
<td>0.24</td>
</tr>
<tr>
<td>Avg.</td>
<td>0.45</td>
<td>0.49</td>
</tr>
</tbody>
</table>

Table 3: Critical values for the two-tailed sign test at $\alpha = 0.05$ and $\alpha = 0.10$. A classifier is significantly better than another if it performs better on at least $w_\alpha$ data sets.
Sign Test – Variants

- Observation: some wins/losses are rather marginal
  - Stricter variant:
    - perform significance test for each dataset (as shown earlier today)
    - regard only significant wins/losses

- In our example:
  - Let’s assume the results on problem 1, 3, 4, 6, 7, 9, 10, 11, 12 are significant

<table>
<thead>
<tr>
<th>Problem</th>
<th>M</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.09</td>
<td>0.11</td>
</tr>
<tr>
<td>2</td>
<td>0.71</td>
<td>0.72</td>
</tr>
<tr>
<td>3</td>
<td>0.77</td>
<td>0.69</td>
</tr>
<tr>
<td>4</td>
<td>0.21</td>
<td>0.44</td>
</tr>
<tr>
<td>5</td>
<td>0.37</td>
<td>0.37</td>
</tr>
<tr>
<td>6</td>
<td>0.85</td>
<td>0.92</td>
</tr>
<tr>
<td>7</td>
<td>0.62</td>
<td>0.65</td>
</tr>
<tr>
<td>8</td>
<td>0.58</td>
<td>0.55</td>
</tr>
<tr>
<td>9</td>
<td>0.79</td>
<td>0.89</td>
</tr>
<tr>
<td>10</td>
<td>0.12</td>
<td>0.16</td>
</tr>
<tr>
<td>11</td>
<td>0.09</td>
<td>0.15</td>
</tr>
<tr>
<td>12</td>
<td>0.19</td>
<td>0.24</td>
</tr>
<tr>
<td>Avg.</td>
<td>0.45</td>
<td>0.49</td>
</tr>
</tbody>
</table>

Table 3: Critical values for the two-tailed sign test at $\alpha = 0.05$ and $\alpha = 0.10$. A classifier is significantly better than another if it performs better on at least $w_\alpha$ data sets.
Wilcoxon Signed-Rank Test

- Observation: some wins/losses are rather marginal
- Wilcoxon Signed-Rank Test
  - takes margins into account

- Approach:
  - rank results by *absolute* difference
  - sum up ranks for positive and negative outcomes
    - best case: all outcomes positive → sum of negative ranks = 0
    - still good case: all negative outcomes are marginal → sum of negative ranks is low

<table>
<thead>
<tr>
<th>Problem</th>
<th>M</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.09</td>
<td>0.11</td>
</tr>
<tr>
<td>2</td>
<td>0.71</td>
<td>0.72</td>
</tr>
<tr>
<td>3</td>
<td>0.77</td>
<td>0.69</td>
</tr>
<tr>
<td>4</td>
<td>0.21</td>
<td>0.44</td>
</tr>
<tr>
<td>5</td>
<td>0.37</td>
<td>0.37</td>
</tr>
<tr>
<td>6</td>
<td>0.85</td>
<td>0.92</td>
</tr>
<tr>
<td>7</td>
<td>0.62</td>
<td>0.65</td>
</tr>
<tr>
<td>8</td>
<td>0.58</td>
<td>0.55</td>
</tr>
<tr>
<td>9</td>
<td>0.79</td>
<td>0.89</td>
</tr>
<tr>
<td>10</td>
<td>0.12</td>
<td>0.16</td>
</tr>
<tr>
<td>11</td>
<td>0.09</td>
<td>0.15</td>
</tr>
<tr>
<td>12</td>
<td>0.19</td>
<td>0.24</td>
</tr>
<tr>
<td>Avg.</td>
<td>0.45</td>
<td>0.49</td>
</tr>
</tbody>
</table>
Wilcoxon Signed-Rank Test

- Computation:
  - sum up $R+$ and $R-$
  - ties are ignored
  - equal ranks are averaged

- $R+=62.5, R-=14.5$
Wilcoxon Signed-Rank Test

- Computation: rank results
  - sum up R- and R+
  - ties are ignored
  - equal ranks are averaged
- \( R^- = 14.5, R^+ = 62.5 \)
- We use the one-tailed test
  - because we want to test if \( M \) is \textit{better} than \( S \)
- \( 14.5 < 17 \)
  - the results are significant

Tests for Comparing Approaches

• Summary
  – Simple z test only reliable for many datasets (>30)
  – Sign test does not distinguish large and small margins
  – Wilcoxon signed-rank test
    • works also for small samples (e.g., half a dozen datasets)
    • considers large and small margins
Ablation Studies

• Often, data mining pipelines are complex
  – different preprocessing approaches
  – adding external data
  – computing extra features
  – …

• Each of those steps may be
  – left out
  – replaced by a simpler baseline

• This is called an ablation study, i.e.,
  – does that change bear a significant advantage?
  – recap: Occam’s razor!
Occam's Razor Revisited

• Let’s rephrase:
  – if you have two models
  – where none is *significantly* better than the other
  – choose the simpler one

• Indicators for simplicity:
  – number of features used
  – number of variables used, e.g.,
    • hidden neurons in an ANN
    • no. of trees in a Random Forest
    • …
Measuring Model Simplicity

• Idea: the less feature the model focuses on, the simpler
  – Not necessarily: the better

• Good models have *both*...
  – ...low test error
  – ...low complexity

Caveats: identifiers, false predictors, ...
Measuring Feature Importance

• Example: random forests

• A feature is more important if...
  – ...it is used in many trees
    Rationale:
    • weighted prediction across trees
    • the more trees it is used in, the higher the influence
  – ...it is used to classify many examples
    Rationale:
    • more predictions are influenced by that attribute
    • i.e., for a single example: higher likelihood of influence
  – ...it leads to a high increase of purity on average
    Rationale:
    • if the purity is *not* increased, the split is rather a coin toss
Measuring Feature Importance

• A feature is more important if…
  – ...it is used in many trees
  – First take:

\[
\text{Importance}(F) = \frac{\text{no. of trees containing } F}{\text{no. of trees}}
\]

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>30</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>False</td>
<td>5</td>
<td>20</td>
<td>5</td>
<td>20</td>
<td></td>
<td>5</td>
<td>20</td>
</tr>
</tbody>
</table>
Measuring Feature Importance

- A feature is more important if...
  - ...it is used to classify many examples
  - First take:

\[
\text{Importance}(F) = \frac{\text{no. of examples classified using } F}{\text{no. examples}}
\]

- In this example tree:
  - Importance(x) = 1.0
  - Importance(y) = 0.6
  - Importance(z) = 0.4
Measuring Feature Importance

• A feature is more important if…
  – ...it leads to a high increase of purity on average
  – First take:
    Change of impurity of node and its split nodes

Importance(\(F\)) = \(\Delta I(t, t_s)\)

– In this example tree:
  • Importance(x) = 0.023
  • Importance(y) = 0.204
  • Importance(z) = 0.087

• gini(A) = 0.5
• gini(B) = 0.486
• gini(C) = 0.469
• gini(D) = 0.245
• gini(E) = 0.320
• gini(F) = 0.444
• gini(G) = 0.320
Measuring Feature Importance

- For example, random forests
- Putting the pieces together:

\[
\text{Importance}(F) = \frac{1}{\text{no. of trees}} \sum_{m=1}^{\text{no. of trees containing } F} \sum_{\text{nodes } n \text{ in tree } m \text{ containing } F} p(n) \Delta I(s_n, n)
\]

- Probability of single example passing this inner node
- Growth in impurity (e.g. Gini, Entropy)

Grows with no. of trees using F
Measuring Feature Importance

• For example, random forests
• Putting the pieces together:

\[
\text{Importance}(F) = \frac{1}{\text{no. of trees}} \sum_{m=1}^{\text{no. of trees containing } F} \sum_{\text{nodes } n \text{ in tree } m \text{ containing } F} p(n) \Delta I(s_n, n)
\]

• In this example:
  – Importance(x) = 1.0 * 0.023 = 0.023
  – Importance(y) = 0.6 * 0.204 = 0.122
  – Importance(z) = 0.4 * 0.087 = 0.035
Back to Model Simplicity

- Left hand side:
  - Accuracy on test set: 0.72
- Right hand side:
  - Accuracy on test set: 0.66

Fewer influential features
Feature Weights and Model Simplicity

- Idea of feature shuffling:
  - If a feature is relevant, assigning random values to it should make the predictions worse
  - Simulation of random, but realistic values: shuffling a column
- This can be applied to *any* model

Back to Model Simplicity

• Left hand side:
  – Accuracy on test set: 0.66

• Right hand side:
  – Accuracy on test set: 0.64

Fewer features with importance >0
Feature Weights and Model Simplicity

• Let’s rephrase:
  – if you have two models
  – where none is significantly better than the other
  – choose the simpler one

• Feature weights
  – Can indicate model simplicity
    (few high weighted features)

• Examples for computation
  – Random Forest, XGBoost: Mean Decrease in Impurity (MDI)
  – General: feature shuffling
LIME Model Explanation

- Idea: in a local area, models are simpler
  - They do not need to account for all the patterns of the data
  - Concentrate on patterns relevant in that area

- Motivation:
  - Try to extract the relevant model for a given data point
  - Hopefully, this is simple enough to interpret

LIME Model Explanation

- How to interpret a “black box” (i.e., uninterpretable) model M?
- Local: for a datapoint p
- Basic idea:
  1) create artificial datapoints \( P(p) \) in vicinity of \( p \)
  2) score each \( p' \) in \( P \) with black box model
  3) learn interpretable model \( M' \)
      \( \rightarrow \) values: \( P \), labels: scores of \( M \)
  4) create prediction for \( p \) using \( M' \)
      or analyze \( M' \) directly

LIME Model Explanation (example)

- Left hand side:
  - Model score on test set: 0.80
- Right hand side:
  - Model score on test set: 0.74
LIME Models for Non-Tabular Data

• Example: text classification
  – Datapoints $P(p)$ are created by changing single *words* in training example

https://towardsdatascience.com/fine-grained-sentiment-analysis-in-python-part-2-2a92fdc0160d
LIME Models for Non-Tabular Data

• Example: image classification
  – Datapoints $P(p)$ are created by changing single pixels in training example

https://www.inovex.de/de/blog/lime-machine-learning-interpretability/
Model Inspection for Improving Model Quality

- **Example: Text Classification**
  - Observation: focus on metadata and stop words

---

Prediction probabilities

<table>
<thead>
<tr>
<th></th>
<th>atheist</th>
<th>christian</th>
</tr>
</thead>
<tbody>
<tr>
<td>atheism</td>
<td>0.58</td>
<td></td>
</tr>
<tr>
<td>christian</td>
<td>0.42</td>
<td></td>
</tr>
</tbody>
</table>

**Text with highlighted words**

From: johncad@triton.unm.edu (jchadwic)
Subject: Another request for Darwin Fish
Organization: University of New Mexico, Albuquerque
Lines: 11

NNTP-Posting-Host: triton.unm.edu

Hello Gang,

There have been some notes recently asking where to obtain the DARWIN fish.
This is the same question I have and I have not seen an answer on the net. If anyone has a contact please post on the net or email me.

https://homes.cs.washington.edu/~marcotcr/blog/lime/
Take Aways

• Results in Data Mining are often reduced to a single number
  – e.g., accuracy, error rate, F1, RMSE
  – result differences are often marginal

• Problem of unseen data
  – we can only guess/approximate the true performance on unseen data
  – makes it hard to select between approaches

• Helpful tools
  – confidence intervals
  – significance tests
  – Occam’s Razor
Take Aways

• Model inspection on global level
  – Model complexity
  – Proxy: feature importance
  – Less complex model → more likely to generalize

• Model inspection on local level
  – Generating explanations for test instances
  – Do they look plausible?
Questions?
Data Mining II
Model Validation