Validation

Sanjay Lall and Stephen Boyd

EE104
Stanford University
Generalization
Generalization

- *generalization* is the ability of a predictor to perform well on unseen data
- can be mathematically analyzed by making probabilistic assumptions, which we won’t discuss in this course
- instead we’ll see some practical methods for assessing generalizability
In-sample and out-of-sample data

- we construct a predictor based on *training data* or *in-sample data*
- we’d like it to work well on *out-of-sample data*
- if it doesn’t we say it *fails to generalize*
we train straight-line model using the 12 (in-sample) blue points, MSE 0.0047

we use this model to predict \( y \) for the 14 (out-of-sample) red points, MSE 0.0051

so, this predictor generalizes
Out-of-sample validation
Out-of-sample validation

- a method to simulate how the predictor will perform on unseen data
- key idea: divide the data into two sets, \textit{train} and \textit{test}

- use the \textit{training set} data to choose (‘train’) the predictor
- use the \textit{test set} or \textit{validation set} data to evaluate the predictor

- this is an honest simulation of how the predictor works on unseen data
- we \textit{hope} that the predictor will work in a similar way on new unseen data
- this hope founded on the assumption that future data ‘looks like’ test data
Out-of-sample validation

- the test set error is what matters, not the training set error
- selection of data for the training/test sets is often random
  (80/20 or 90/10 are common splits)
- we expect the test error to be a little bigger than the training error
- if the test error is much greater than training error, the predictor is overfit
  (but if the test error is acceptable, this can still be useful)
## Interpreting validation results

<table>
<thead>
<tr>
<th></th>
<th>small training error</th>
<th>large training error</th>
</tr>
</thead>
<tbody>
<tr>
<td>small test error</td>
<td>generalizes, performs well</td>
<td>possible (luck, or fraud?)</td>
</tr>
<tr>
<td>large test error</td>
<td>fails to generalize</td>
<td>generalizes, but performs poorly</td>
</tr>
</tbody>
</table>
Choosing a model
Choosing among candidate models

- validation is a good method to choose among candidate models
- typically we choose model among candidates with smallest test error
- in some cases, might accept a bit larger test error in favor of a ‘simpler’ model (more on this later)
Example: Diabetes

- 10 explanatory variables (age, bmi, ...)
- data from 442 individuals
- use half for training, half for validation (50-50 split)
## Example: Diabetes

<table>
<thead>
<tr>
<th>features</th>
<th>train loss</th>
<th>test loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>2640</td>
<td>3224</td>
</tr>
<tr>
<td>S5 and BMI</td>
<td>3004</td>
<td>3453</td>
</tr>
<tr>
<td>S5</td>
<td>3869</td>
<td>4227</td>
</tr>
<tr>
<td>BMI</td>
<td>3540</td>
<td>4277</td>
</tr>
<tr>
<td>S4 and S3</td>
<td>4251</td>
<td>5302</td>
</tr>
<tr>
<td>S4</td>
<td>4278</td>
<td>5409</td>
</tr>
<tr>
<td>S3</td>
<td>4607</td>
<td>5419</td>
</tr>
<tr>
<td>none</td>
<td>5524</td>
<td>6352</td>
</tr>
</tbody>
</table>

- Test loss gives a method of selecting features
- Data indicates that using only 2 features, S5 and BMI, would predict diabetes almost as well as using all 10 features
- Combining S4 and S3 doesn’t buy much; combining S5 and BMI much better
Overfitting
Overfitting

- we have a family of models
- we might choose a model that fits the training data very closely
- but often this leads to poorly fitting the test data
- called overfitting
Example: Polynomial fit

- raw data is scalar \( u \in \mathbb{R} \)
- we use \textit{polynomial features}

\[
x = \phi(u) = \begin{bmatrix}
1 \\
u \\
u^2 \\
\vdots \\
u^{d-1}
\end{bmatrix}
\]

and linear model \( g(x) = \theta^T x \)

- prediction model is polynomial of \( u \) of degree \( d - 1 \):

\[
\hat{y} = g(x) = \theta_1 + \theta_2 u + \cdots + \theta_d u^{d-1}
\]

- choose \( \theta \) by ERM with square loss \( l^{\text{sqr}}(\hat{y}, y) = (\hat{y} - y)^2 \)
Example: Polynomial fit

- $n = 60$ data points
- predictor for $d = 6$, $d = 12$, $d = 14$
Choosing degree by validation

- split 60 data points into 48 train and 12 test points
- plot suggests best choice of degree is 5
- can now use degree 5 fit on all data
Cross validation
Cross validation

- an extension of out-of-sample validation
  - divide the data into $k$ folds
  - for each $i$, fit model on all data but fold $i$
  - evaluate model on fold $i$
  - use average test error, across the folds, to judge the method

- can give some idea of the variability of the test error

- can assess *stability* of the modeling method by looking at model parameters found in each fold (are they similar? very different?)
Example: Cross validation

50 data points, artificial data

<table>
<thead>
<tr>
<th>fold</th>
<th>training loss</th>
<th>test loss</th>
<th>$\theta_1$</th>
<th>$\theta_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.028</td>
<td>0.030</td>
<td>-0.016810</td>
<td>0.9874</td>
</tr>
<tr>
<td>2</td>
<td>0.026</td>
<td>0.036</td>
<td>0.005917</td>
<td>0.9822</td>
</tr>
<tr>
<td>3</td>
<td>0.030</td>
<td>0.023</td>
<td>0.008961</td>
<td>1.0010</td>
</tr>
<tr>
<td>4</td>
<td>0.028</td>
<td>0.031</td>
<td>0.004135</td>
<td>0.9859</td>
</tr>
<tr>
<td>5</td>
<td>0.028</td>
<td>0.029</td>
<td>0.000844</td>
<td>0.9742</td>
</tr>
</tbody>
</table>
And to be even more confident . . .

- split data into train:test (say, 80:20) *randomly*
- develop model from training data
- evaluate on test data
- repeat above for many different random splits into train:test
- look at histogram of test errors to judge the method
- called *repeated train/validation*
Example: Repeated train/validation

- 1000 experiments
- diabetes data, with BMI and S5 features
- mean loss: 3258