Mathematical Tools for Neural and Cognitive Science

Fall semester, 2021

Section 6

Model fitting: comparison, selection and regularization
Taxonomy of model-fitting errors

- Unexplainable variability (due to finite/noisy measurements)
- Overfitting (too many params, not enough data)
- Optimization failures (e.g., local minima)
- Model failures (what you’d really like to know)
Optimization...

- Closed-form, and unique
- Iterative descent, unique
- Convex
- Smooth ($C_2$)
- Heuristics, exhaustive search, (pain & suffering)
- Iterative descent, possibly non-unique (local minima)
Model Comparison

• If models are optimized according to some objective, it is natural to compare them based on the value of that objective…
  
  - for least squares regression, compare the residual squared error of two models (with different regressors).
  
  - for ML estimates, compute the likelihood (or log likelihood) ratio, and compare to 1 (or zero).
  
  - for MAP estimates, common to compute the posterior ratio (a.k.a. the Bayes factor)

• Problem: evaluating the objective with the same data used to optimize the model leads to over-fitting! We really want to predict error on non-training data…
How do we avoid overfitting (i.e., concluding that $M=7$ is “best”)?
Comparing models’ predictive performance

Option 1: Include a penalty for number of parameters:

For an ML estimate:

\[ \hat{\theta} = \arg \min_{\theta} \left[ - \ln p(d|\theta) \right] \]

a. Akaike information criterion (AIC) [Akaike, 1974]

\[ E_{\text{AIC}}(\vec{d}, \hat{\theta}) = 2 \dim(\hat{\theta}) - 2 \ln p(\vec{d}|\hat{\theta}) \]

b. Bayesian information criterion (BIC) [Schwartz, 1978]

\[ E_{\text{BIC}}(\vec{d}, \hat{\theta}) = \dim(\hat{\theta}) \ln \left[ \dim(\vec{d}) \right] - 2 \ln p(\vec{d}|\hat{\theta}) \]

valid when \( \dim(\vec{d}) \gg \dim(\hat{\theta}) \)

Option 2: Cross-validation: partition data into two subsets, fit parameters to “training” subset, evaluate objective on “test” subset.
Cross-validation

A resampling method for estimating predictive error of a model. Widely used to identify/avoid over-fitting, and to provide a fair comparison of models.

1. Randomly partition data into a “training” set, and a “test” set.
2. Fit model to training set. Measure error on test set.
3. Repeat (many times).
4. Choose model that minimizes the average cross-validated (“test”) error.

Using cross-validation to select the degree of a polynomial model:
Ridge regression
(a.k.a. Tikhonov regularization)

Ordinary least squares regression:
\[
\arg \min_{\beta} \| \tilde{y} - X\beta \|^2
\]

“Regularized” least squares regression:
\[
\arg \min_{\beta} \| \tilde{y} - X\beta \|^2 + \lambda \| \beta \|^2
\]

Equivalent formulation: MAP estimate, assuming Gaussian likelihood & prior!

\[
\hat{\beta}_{\text{ridge}} = (X^TX + \lambda I)^{-1}X^T\tilde{y}
\]

Choose lambda by cross-validation:
Ridge Regression trades off bias and variance:

![Graph showing the trade-off between bias and variance in Ridge Regression](http://www.stat.cmu.edu/~ryantibs/datamining/)
**$L_1$ regularization**

(a.k.a. “least absolute shrinkage and selection operator” - LASSO)

$$\arg \min_\beta \|\tilde{y} - \tilde{x}/\beta\|^2 + \lambda|\beta|$$

Assuming $\|\tilde{x}\|^2 = 1$,

*[derivation on board]*

MAP interpretation:
Gaussian noise,
“Laplacian” prior
$L_2$ regularization

$L_1$ regularization

\[ \beta_1 \]

\[ \beta_2 \]

\[ \lambda = 0 \]

\[ \lambda = \infty \]

\[ \text{other } \lambda \]
LASSO vs. ridge regression

Table 2.1 Crime data: Crime rate and five predictors, for N = 50 U.S. cities.

<table>
<thead>
<tr>
<th>city</th>
<th>funding</th>
<th>hs</th>
<th>not-hs</th>
<th>college</th>
<th>college4</th>
<th>crime rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40</td>
<td>74</td>
<td>11</td>
<td>31</td>
<td>20</td>
<td>478</td>
</tr>
<tr>
<td>2</td>
<td>32</td>
<td>72</td>
<td>11</td>
<td>43</td>
<td>18</td>
<td>494</td>
</tr>
<tr>
<td>3</td>
<td>57</td>
<td>70</td>
<td>18</td>
<td>16</td>
<td>16</td>
<td>643</td>
</tr>
<tr>
<td>4</td>
<td>31</td>
<td>71</td>
<td>11</td>
<td>25</td>
<td>19</td>
<td>341</td>
</tr>
<tr>
<td>5</td>
<td>67</td>
<td>72</td>
<td>9</td>
<td>29</td>
<td>24</td>
<td>773</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>50</td>
<td>66</td>
<td>67</td>
<td>26</td>
<td>18</td>
<td>16</td>
<td>940</td>
</tr>
</tbody>
</table>

Lasso and ridge regression coefficient paths for the lasso and ridge regression. The residual sum profiles, but are not equal to zero except at the left end. Figure 2.2 contrasts the ridge profiles in the right panel have roughly the same shape as the lasso to the geometry that underlies the two constraints used in the lasso and ridge regression. The residual sum of squares function. The horizontal axis has been scaled so that the maximal bound, corresponding to the least-squares estimates exactly zero and hence the corresponding predictor(s) would be excluded from the right, where it has no effect.

As an example of the lasso, let us consider the data given in Table 2.1, taken from Thomas (1990). The outcome is the total overall reported crime rate per one million residents in 50 U.S. cities. There are five predictors: annual police funding in dollars per resident, percent of people 25 years and older with four years of college, percent of people 25 years and older with at least four years of college. This small example is for illustration only, but helps to demonstrate the nature of the algorithms for finding its solutions. More details are given in Exercises (2.3) and (2.4).

Table 2.1 Crime data: Crime rate and five predictors, for N = 50 U.S. cities.

<table>
<thead>
<tr>
<th>city</th>
<th>funding</th>
<th>hs</th>
<th>not-hs</th>
<th>college</th>
<th>college4</th>
<th>crime rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40</td>
<td>74</td>
<td>11</td>
<td>31</td>
<td>20</td>
<td>478</td>
</tr>
<tr>
<td>2</td>
<td>32</td>
<td>72</td>
<td>11</td>
<td>43</td>
<td>18</td>
<td>494</td>
</tr>
<tr>
<td>3</td>
<td>57</td>
<td>70</td>
<td>18</td>
<td>16</td>
<td>16</td>
<td>643</td>
</tr>
<tr>
<td>4</td>
<td>31</td>
<td>71</td>
<td>11</td>
<td>25</td>
<td>19</td>
<td>341</td>
</tr>
<tr>
<td>5</td>
<td>67</td>
<td>72</td>
<td>9</td>
<td>29</td>
<td>24</td>
<td>773</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>50</td>
<td>66</td>
<td>67</td>
<td>26</td>
<td>18</td>
<td>16</td>
<td>940</td>
</tr>
</tbody>
</table>

[From Hastie, Tibshirani, Wainwright 2015]
The “Relaxed LASSO”

To reduce bias, re-solve for non-zero coefficients after eliminating unused regressors.
Clustering

• K-Means (Lloyd, 1957)

• “Soft-assignment” version of K-means
  (a form of Expectation-Maximization - EM)

• In general, alternate between:
  1) Estimating cluster assignments
  2) Estimating cluster parameters

• Coordinate descent: converges to (possibly local) minimum

• Need to choose K (number of clusters) - cross-validation!
K-Means algorithm - alternate between two steps:

- Estimating cluster assignments: given class centers, assign each point to closest one.

- Estimating cluster parameters: given assignments, re-estimate the centroid of each cluster.

Soap bubbles:
$K$-means example

Here $X_i \in \mathbb{R}^2$, $n = 300$, and $K = 3$

[from R. Tibshirani, 2013]
Warning: Initialization matters (due to local minima) …

Three solutions obtained with different random starting points:
K-means failures

Non-convex/non-round-shaped clusters

Clusters with different densities

Picture courtesy: Christof Monz (Queen Mary, Univ. of London)
ML for discrete mixture of Gaussians: soft K-means

\[
p(\mathbf{x}_n | a_{nk}, \mathbf{\mu}_k, \Lambda_k) \propto \sum_k \frac{a_{nk}}{\sqrt{|\Lambda_k|}} e^{-(\mathbf{x}_n - \mathbf{\mu}_k)^T \Lambda_k^{-1} (\mathbf{x}_n - \mathbf{\mu}_k)/2}
\]

\[a_{nk} = \text{assignment probability}\]

\[\{\mathbf{\mu}_k, \Lambda_k\} = \text{mean/covariance of class } k\]

Intuition: alternate between maximizing these two sets of variables ("coordinate descent")

Essentially, a version of K-means with "soft" (i.e., continuous, as opposed to binary) assignments!
Different cluster analysis results on "mouse" data set:

Original Data

k-Means Clustering

EM Clustering
Application to neural “spike sorting”

Standard solution:

1. Threshold to find segments containing spikes
2. Reduce dimensionality of segments using PCA
3. Identify spikes using clustering (e.g., K-means)

Note: Fails for overlapping spikes!
Failures of clustering for near-synchronous spikes

synchronous spiking

PC 1 projection

[In Paper Text]

[Pillow et. al. 2013]
Simulated data [Quiroga et. al. 2004]

clustering (K-means)

[Quiroga et. al. 2004]

CBP

[Ekanadham et al, 2014]