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...

- Heuristics, exhaustive search, (pain & suffering)
- Iterative descent, possibly non-unique (local minima)
- Iterative descent, unique
- Closed-form, and unique
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…

Comparing models’ predictive performance

Option 1: Include a penalty for number of parameters:
For an ML estimate: \( \hat{\theta} = \arg \min_{\theta} \left[ -\ln p(\tilde{d}|\theta) \right] \)

a. Akaike information criterion (AIC) [Akaike, 1974]
\[ E_{AIC}(\tilde{d}, \hat{\theta}) = 2 \dim(\hat{\theta}) - 2 \ln p(\tilde{d}|\hat{\theta}) \]
b. Bayesian information criterion (BIC) [Schwartz, 1978]
\[ E_{BIC}(\tilde{d}, \hat{\theta}) = \dim(\hat{\theta}) \ln \left[ \dim(\tilde{d}) \right] - 2 \ln p(\tilde{d}|\hat{\theta}) \]
valid when \( \dim(\tilde{d}) \gg \dim(\hat{\theta}) \)

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. $L_2$ regularization)

Ordinary least squares regression:
$$\arg \min_{\hat{\beta}} ||\hat{y} - X\hat{\beta}||^2$$

"Regularized" least squares regression:
$$\arg \min_{\hat{\beta}} ||\hat{y} - X\hat{\beta}||^2 + \lambda ||\hat{\beta}||^2$$

Equivalent formulation: MAP estimate, assuming Gaussian likelihood & prior!
$$\hat{\beta}_{\text{ridge}} = (X^TX + \lambda I)^{-1}X^T\hat{y}$$

Choose lambda by cross-validation:

Ridge Regression trades off bias and variance:
\textbf{L}_1 \text{ regularization}

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

argin _\beta \min ||\hat{y} - \hat{x}\beta||^2 + \lambda||\beta||

Assuming \(||\hat{x}||^2 = 1\),

[derivation on board]

MAP interpretation:
Gaussian noise,
“Laplacian” prior

\[
\begin{align*}
\hat{\beta}_{OLS} &= 0 \\
\hat{\beta}_{LASSO} &= \lambda/2 \\
\hat{\beta}_{ridge} &= \lambda/2
\end{align*}
\]

\textbf{L}_2 \text{ regularization}

\textbf{L}_1 \text{ regularization}

\textit{level surfaces of regularizer}

\textbf{LASSO vs. ridge regression}

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

\begin{tabular}{ccccccc}
\hline
\text{city} & \text{funding} & \text{hs} & \text{artic} & \text{college} & \text{college} & \text{crime rate} \\
\hline
1 & 30 & 73 & 11 & 25 & 18 & 184 \\
2 & 32 & 73 & 11 & 25 & 18 & 184 \\
3 & 37 & 76 & 19 & 15 & 16 & 163 \\
4 & 31 & 71 & 11 & 25 & 19 & 301 \\
5 & 47 & 72 & 9 & 20 & 24 & 373 \\
\hline
\end{tabular}

\textit{best (cross-validated) lambda}

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

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

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:

1. Estimate cluster assignments: given class centers, assign each point to closest one.

2. Estimating cluster parameters: given assignments, re-estimate the centroid of each cluster.
$K$-means example

$N = 300$, and $K = 3$

[Diagrams showing $K$-means examples]

K-means optimization failures

Initialization matters (due to local minima) …

Three solutions obtained with different random starting points:

[Diagrams showing K-means optimization failures]

K-means systematic failures

Non-convex/non-round-shaped clusters

Clustering with different densities

[Diagrams showing K-means systematic failures]

from R. Tibshirani, 2013
ML for discrete mixture of Gaussians: soft K-means

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

- \( a_{nk} \): assignment probability
- \( \{ \tilde{\mu}_k, \Lambda_k \} \): 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!

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

[Image of a diagram showing synchronous spiking and PC 1 projection]

[Pillow et. al. 2013]