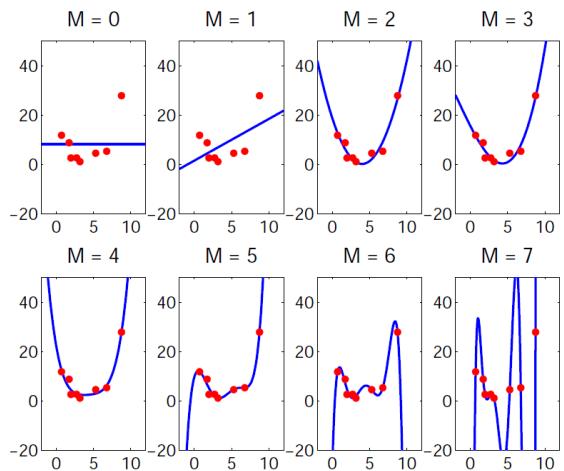


# Mathematical Tools for Neural and Cognitive Science

Fall semester, 2023

## Section 6

### Model fitting: comparison, selection and regularization

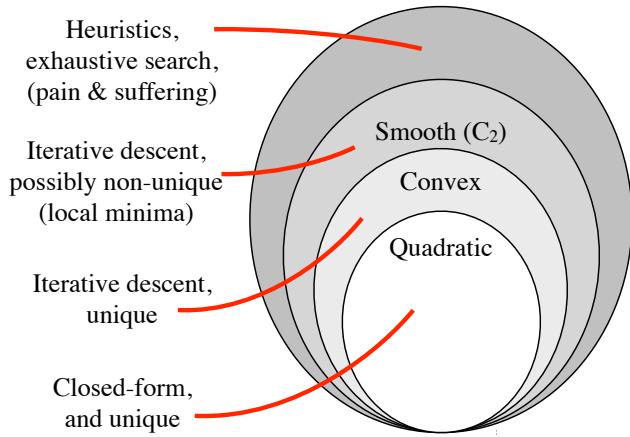


How do we avoid overfitting (i.e., concluding that M=7 is “best”)?

## 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...



## 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...

## Bayesian Model Comparison

- Eg: Is the coin fair? Compared to what?
- Consider two models:  $M_1 : p = 0.5$     $M_2 : p = 0.6$

$$p(M_k | D) = \frac{p(D | M_k)P(M_k)}{p(D)}$$

Compare their posterior ratio (a.k.a. the *Bayes factor*):

$$\frac{p(M_1 | D)}{p(M_2 | D)} = \frac{p(D | M_1)P(M_1)}{p(D | M_2)P(M_2)}$$

## Comparing models' predictive performance

Option 1: Include a penalty for number of parameters:

$$\text{For an ML estimate: } \hat{\theta} = \arg \min_{\theta} \left[ -\ln p(\vec{d}|\theta) \right]$$

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

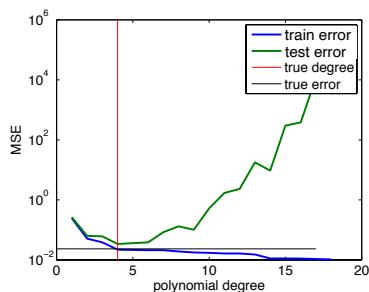
(simulated test of generalization to additional data)

## 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:

$$\arg \min_{\beta} \|\vec{y} - X\vec{\beta}\|^2$$

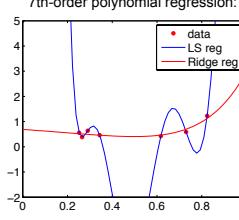
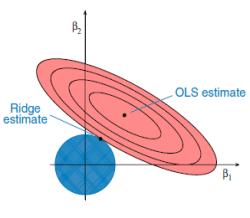
“Regularized” least squares regression:

$$\arg \min_{\beta} \|\vec{y} - X\vec{\beta}\|^2 + \lambda \|\vec{\beta}\|^2$$

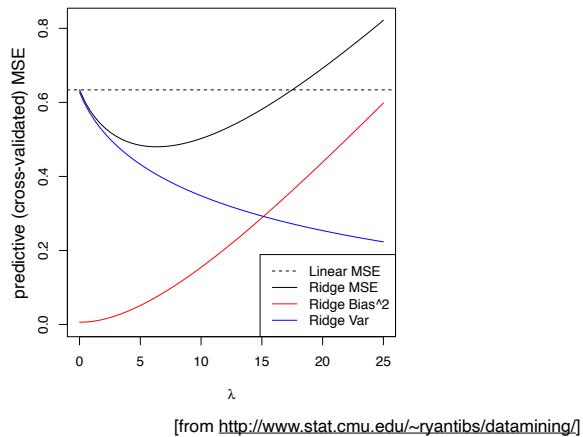
Equivalent formulation: MAP estimate,  
assuming Gaussian likelihood & prior!

$$\hat{\beta}_{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T \vec{y}$$

Choose lambda by cross-validation:



Ridge Regression trades off bias and variance:



### $L_1$ regularization

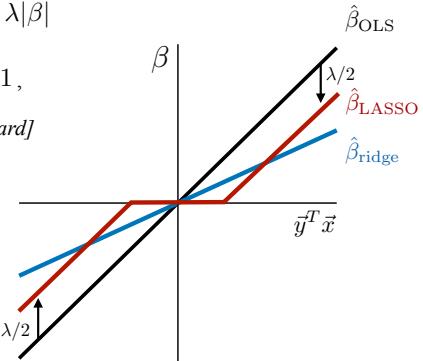
(a.k.a. “least absolute shrinkage and selection operator” - LASSO)

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

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

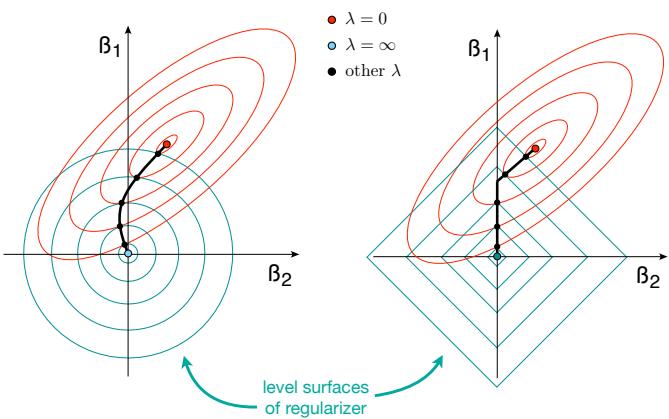
[derivation on board]

MAP interpretation:  
Gaussian noise,  
“Laplacian” prior



### $L_2$ regularization

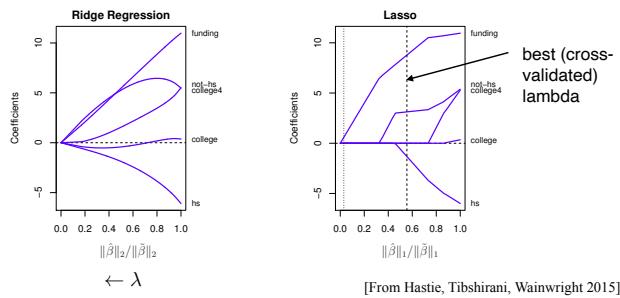
### $L_1$ regularization



## LASSO vs. ridge regression

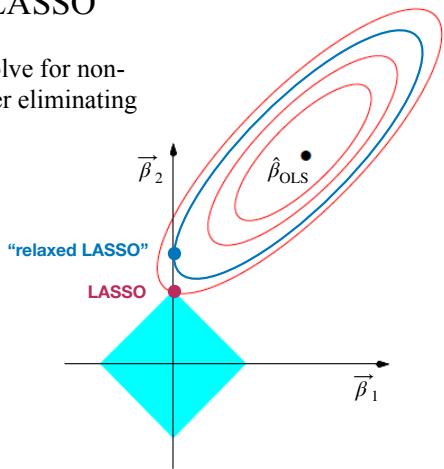
**Table 2.1** Crime data: Crime rate and five predictors, for  $N = 50$  U.S. cities.

city	funding	hs	not-hs	college	college4	crime rate
1	40	74	11	31	20	478
2	32	72	11	43	18	494
3	57	70	18	16	16	643
4	31	71	11	25	19	341
5	67	72	9	29	24	773
⋮	⋮	⋮	⋮	⋮	⋮	⋮
50	66	67	26	18	16	940



## The “Relaxed LASSO”

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

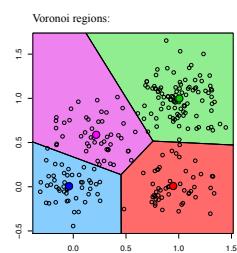


## Clustering

- 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:

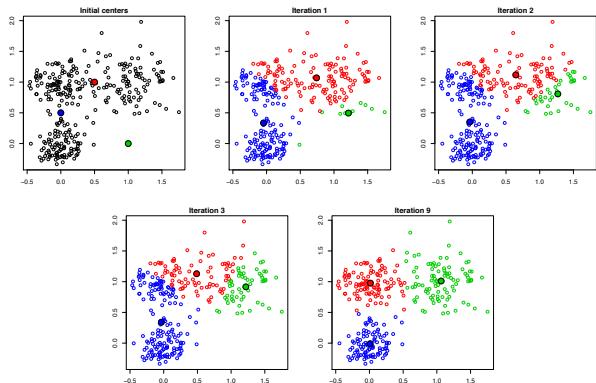
1. Estimate cluster assignments: given class centers, assign each point to closest one.



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

### K-means example

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

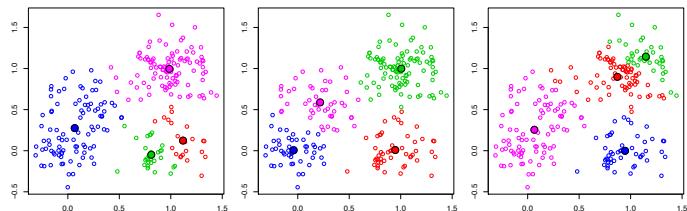


[from R. Tibshirani, 2013]

### K-means optimization failures

Initialization matters (due to local minima) ...

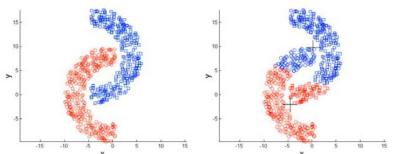
Three solutions obtained with different random starting points:



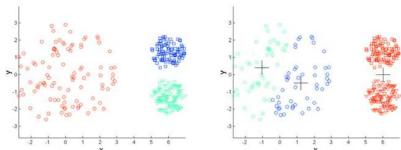
[from R. Tibshirani, 2013]

## K-means systematic 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

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

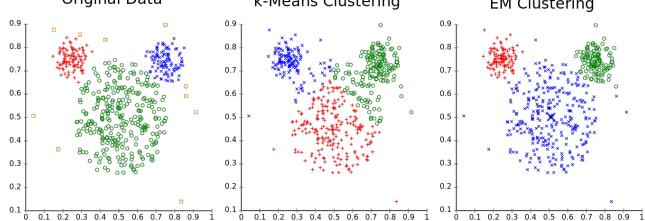
$a_{nk}$  = assignment probability

$\{\vec{\mu}_k, \Lambda_k\}$  = mean/covariance of class  $k$

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

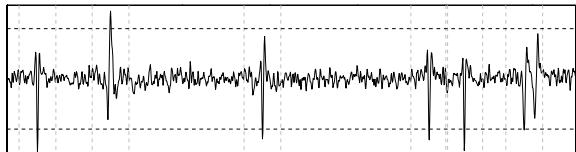
Essentially, a version of K-means with “soft” (i.e., continuous, as opposed to binary) assignments!

Different cluster analysis results on "mouse" data set:



[wikipedia]

## Application to neural “spike sorting”



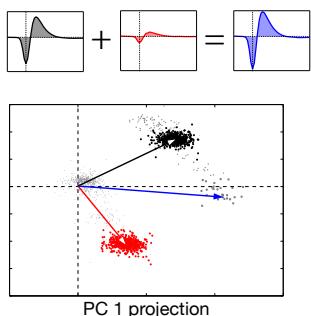
Standard solution:

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



[Pillow et. al. 2013]