Linear Regression

ddebarr@uw.edu

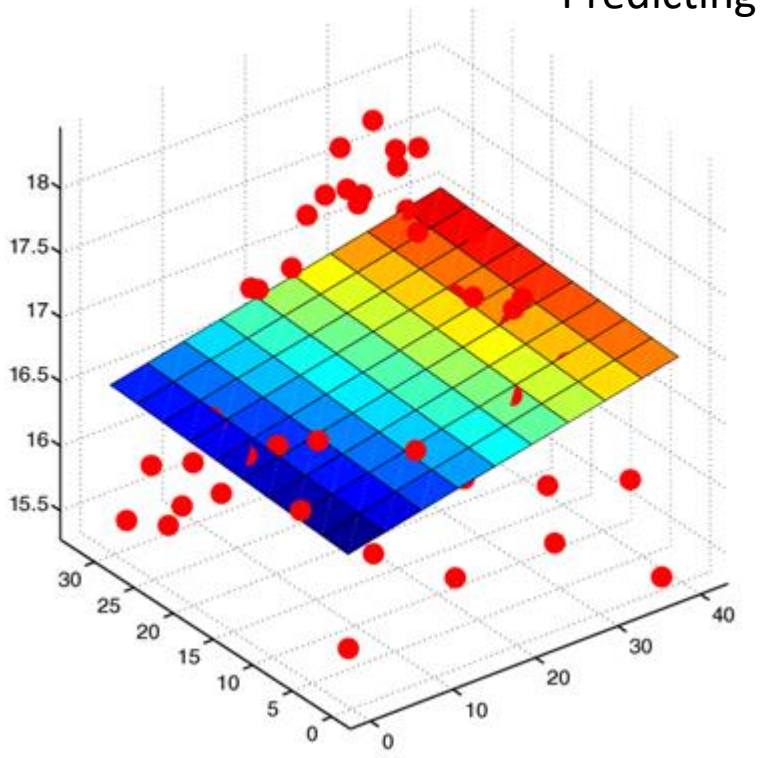
2016-05-19

Agenda

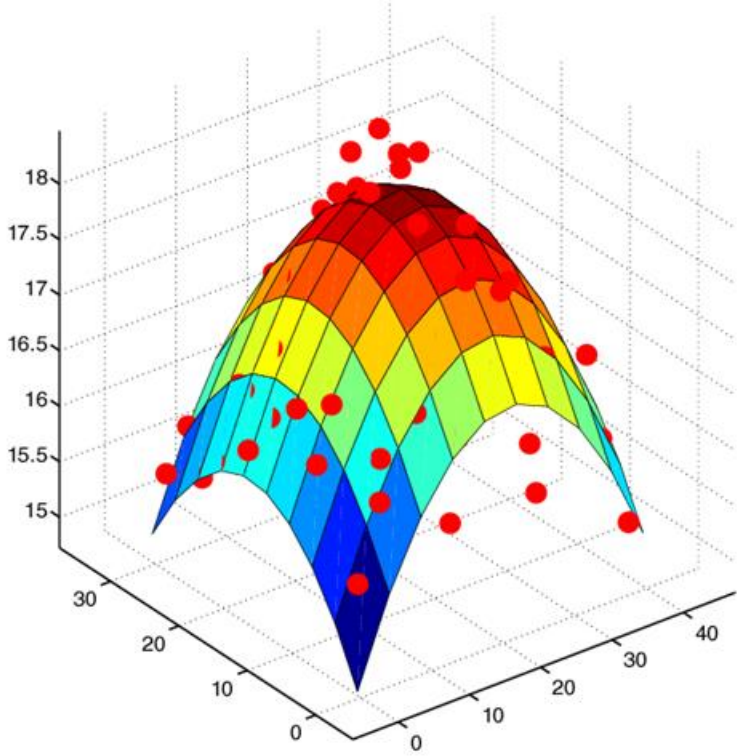
- Model Specification
- Maximum Likelihood Estimation [least squares]
- Robust Linear Regression
- Ridge Regression
- Bayesian Linear Regression

Fitted Plane versus Quadratic Form

Predicting temperature based on position in a room



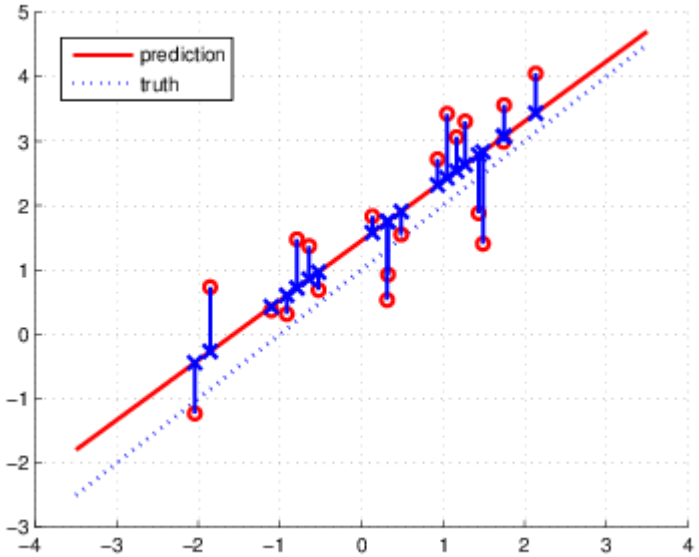
$$\hat{f}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$$



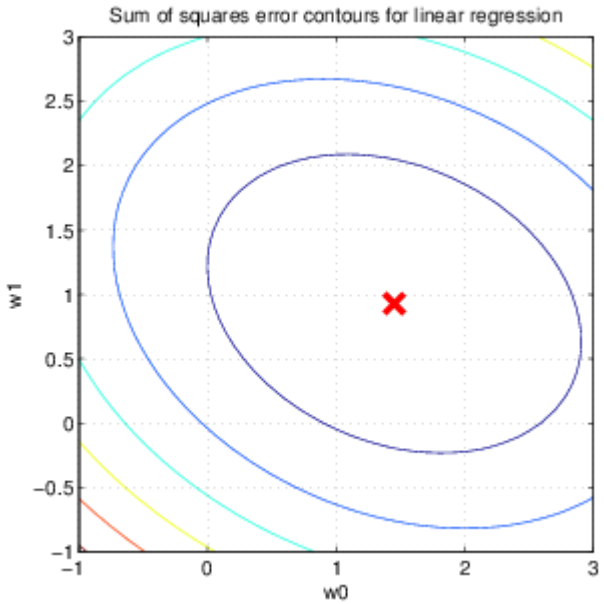
$$\hat{f}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2$$

Both are considered to be linear models

Residuals and Error Contours for Weights



Visualization of residuals (errors)



Contours for weight space

Derivation of the MLE

The Negative Log Likelihood is proportional to the SSE

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\mathbf{w}$$

$$SSE = (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$SSE = (\mathbf{y}^T - \mathbf{w}^T \mathbf{X}^T)(\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$SSE = \mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\mathbf{w} - \mathbf{w}^T\mathbf{X}^T\mathbf{y} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}$$

$$SSE = \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\mathbf{w} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}$$

... so ...

$$\frac{\partial SSE}{\partial \mathbf{w}} = 0 - 2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\mathbf{w}$$

... setting the gradient equal to 0 and solving for $\mathbf{w}$ ...

$$-2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\mathbf{w} = 0$$

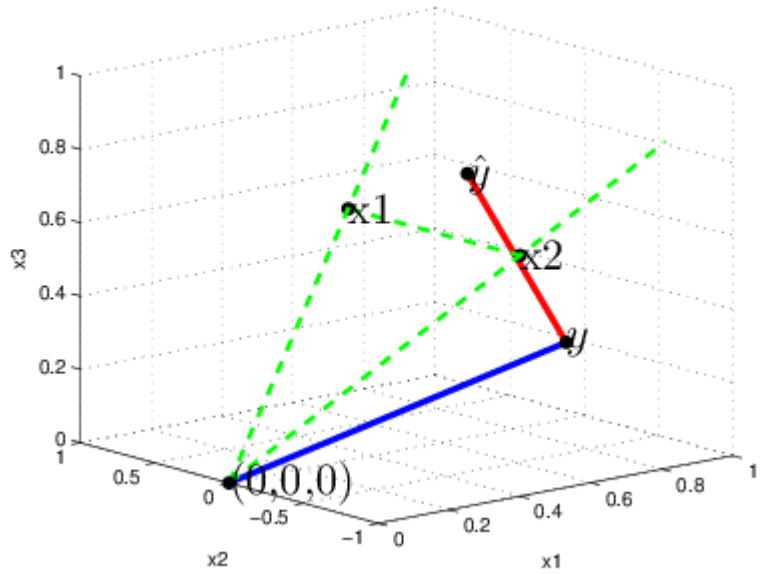
$$2\mathbf{X}^T\mathbf{X}\mathbf{w} = 2\mathbf{X}^T\mathbf{y}$$

$$\mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y}$$

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

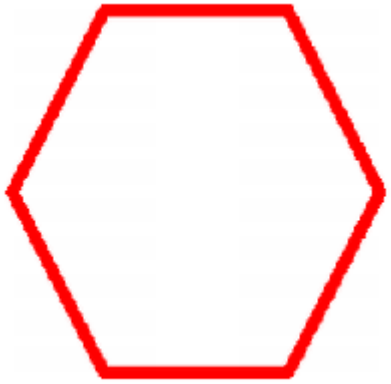
Ref: Equation 7.16

Graphical Interpretation of Least Squares




```
Xnorm =
    0.5774    0.5774
    0.5774   -0.5774
    0.5774    0.5774
ynorm =
    0.9784
    0.0674
    0.1954
wHatNorm =
    0.5666
    0.4499
```

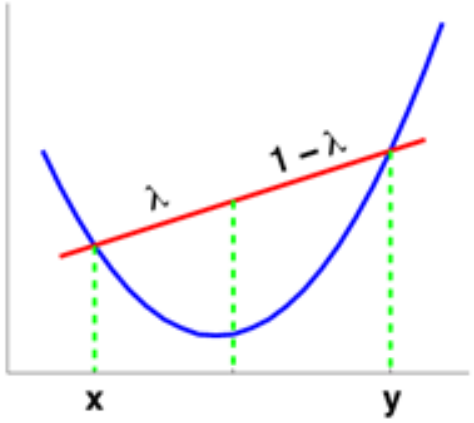
Convex versus Non-Convex Sets



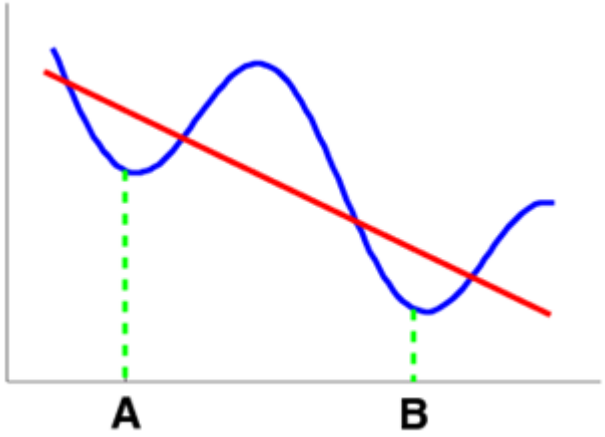
Convex: all points in a line between member points are also member points

Non-Convex

Convex versus Non-Convex Functions



Convex function: any chord between two points of the function lies above the function

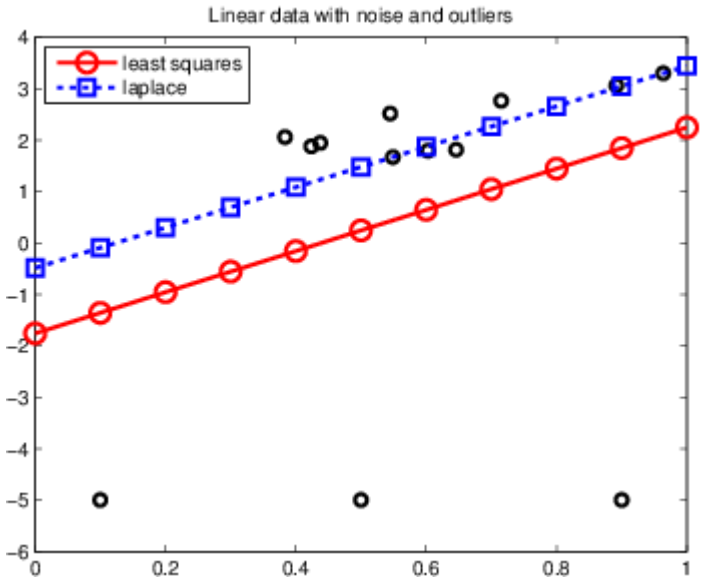


Non-Convex

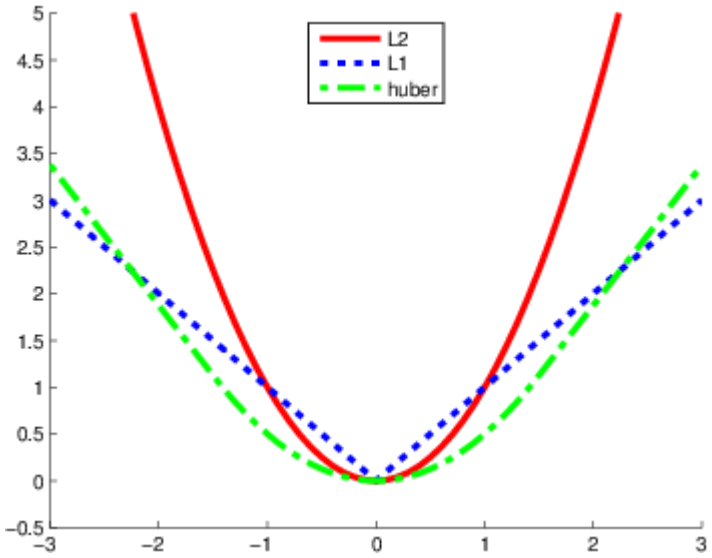
Magic: a convex function has a unique global minimum

Robust Linear Regression

Example Regression Problem



Graph of Loss Functions

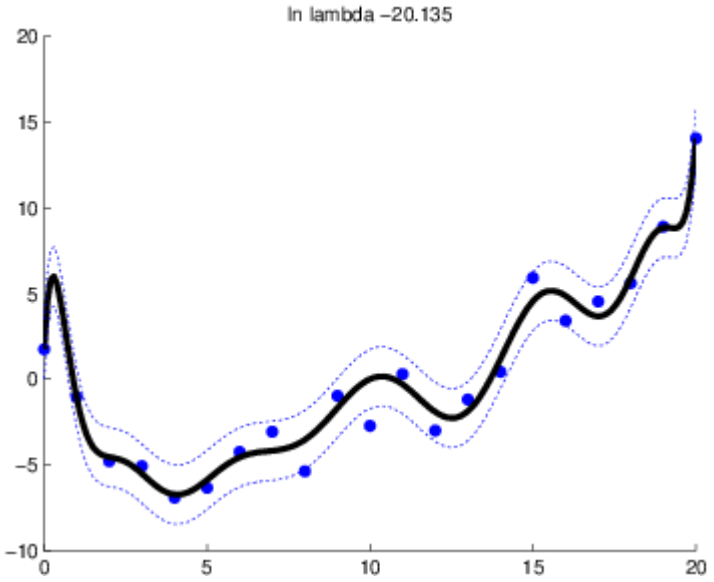


Ordinary least squares: Gaussian “l2” loss
Robust regression: Laplacian “l1” loss [solved via linear programming]

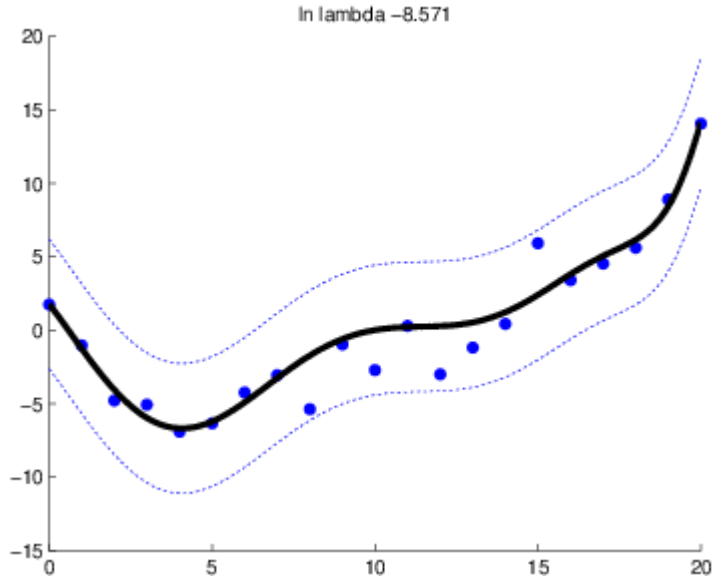
Ridge Regression

$$\hat{\mathbf{w}}_{ridge} = (\lambda \mathbf{I}_D + \mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

polynomial regression example



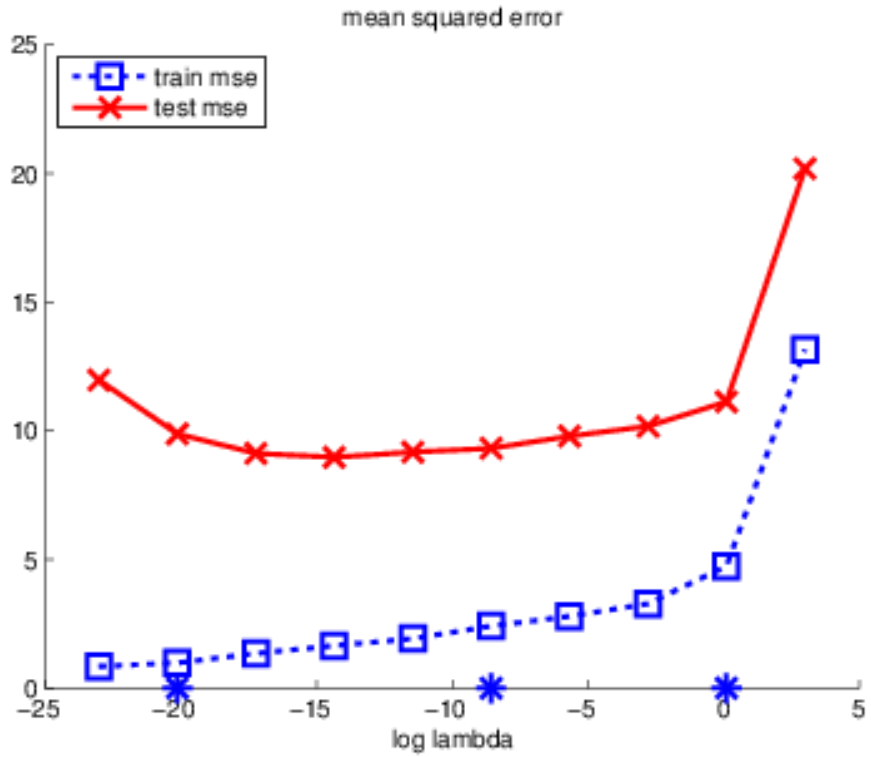
Higher variance

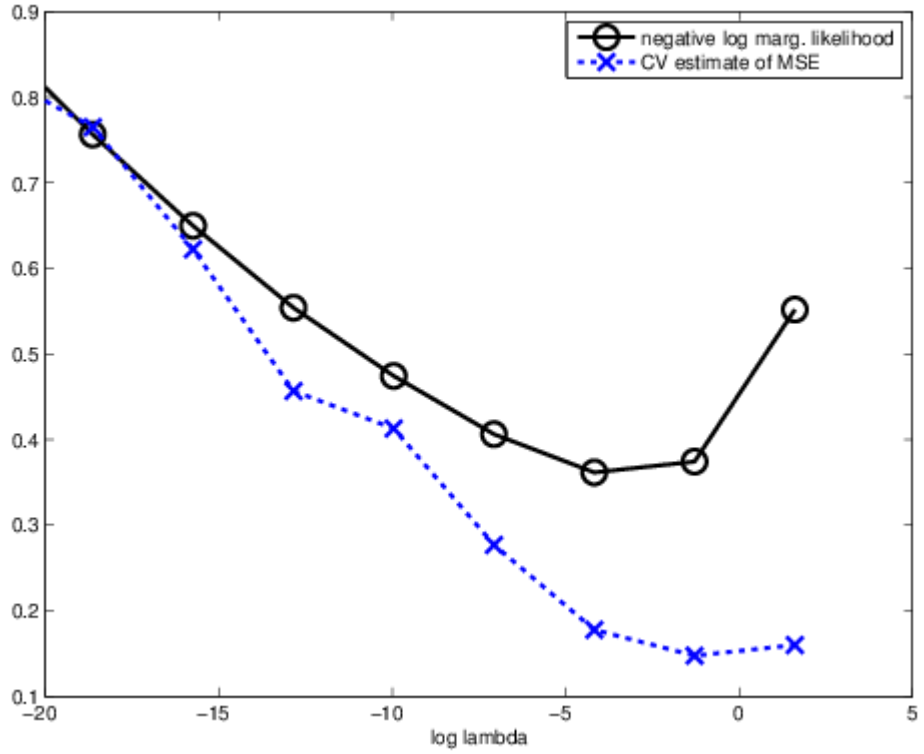


Higher bias

Ridge regression helps avoid overfitting by minimizing $\frac{1}{N}\sum_{i=1}^{N}(y_i - (w_0 + \mathbf{w}^T\mathbf{x}_i))^2 + \lambda||\mathbf{w}||_2^2$

5-Fold Cross Validation in Action



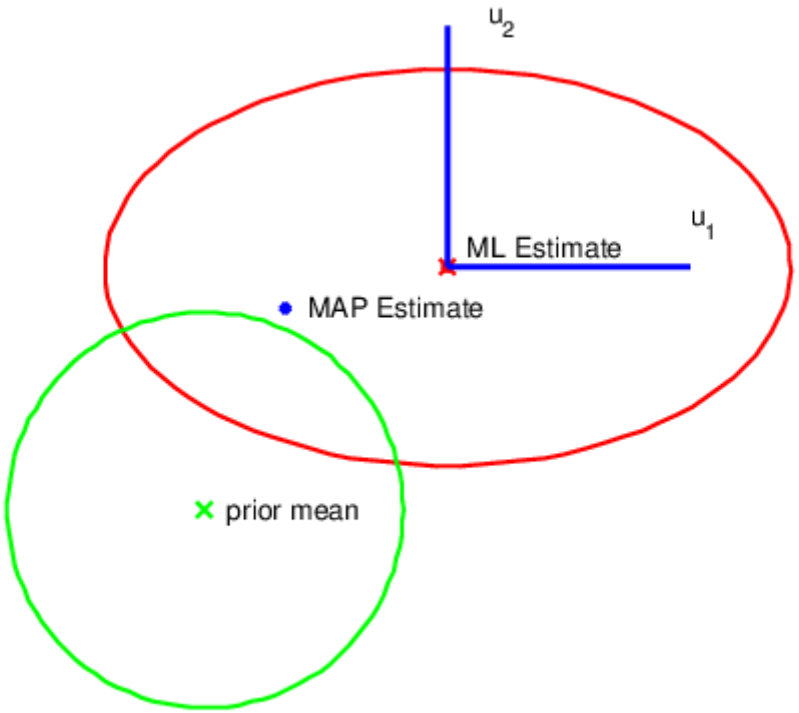


train has 21 observations
test has 201 observations

Results are similar for negative marginal log likelihood and cross validation

Recommendation: avoid using the training data to evaluate fit

Geometry of Ridge Regression

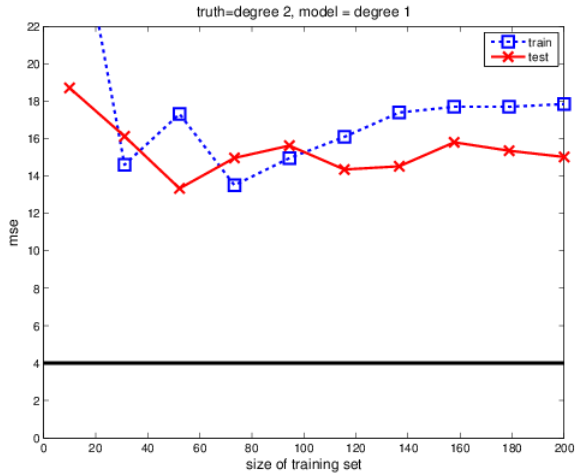


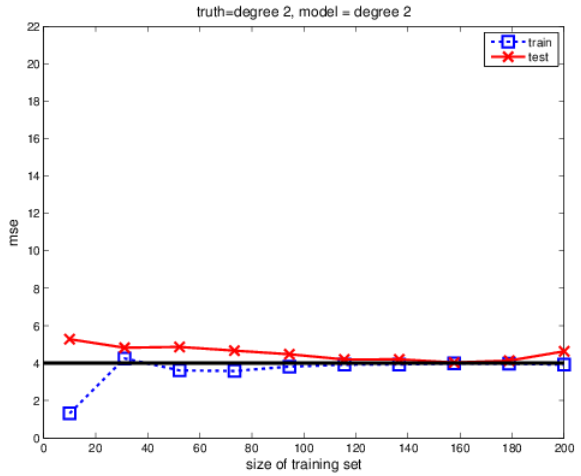
Minimizing Gaussian loss with a penalty on the sum of squared weights means we have a preference for smaller weights

Example Learning Curves

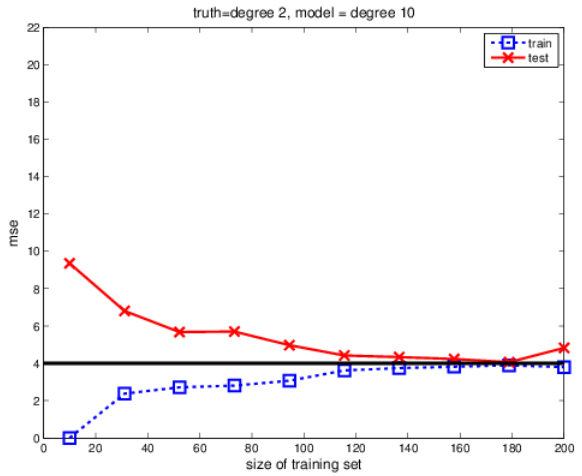
Structural error

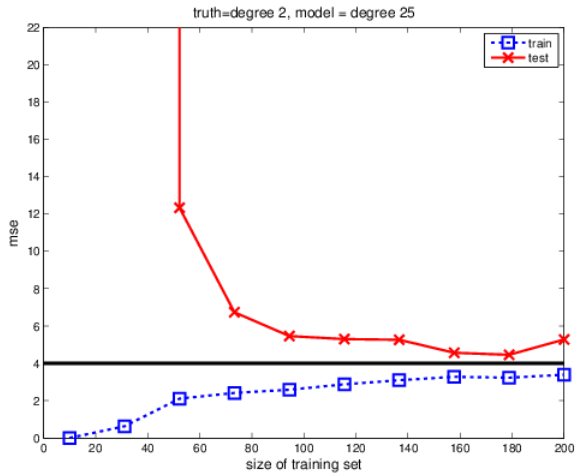
Noise floor





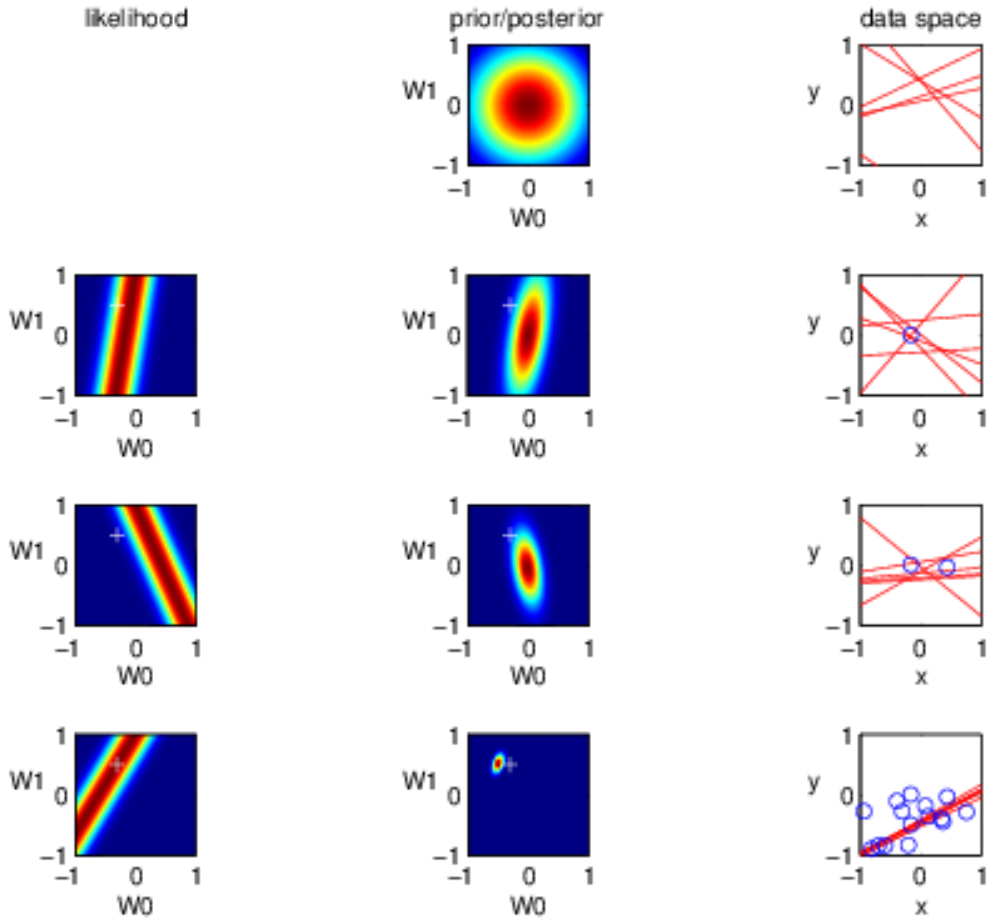
Correct model has quickest convergence





Big data helps reduce overfitting

Sequential Updates



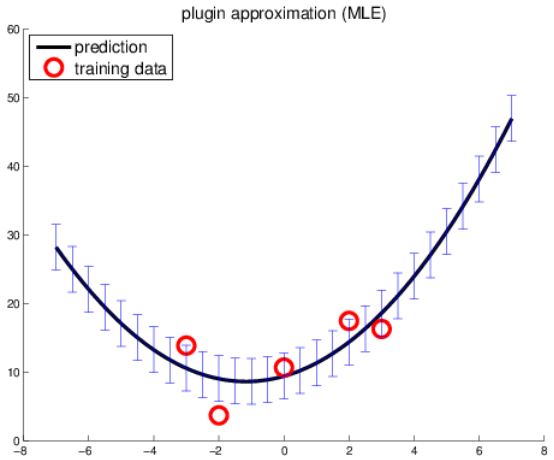
Before we see data

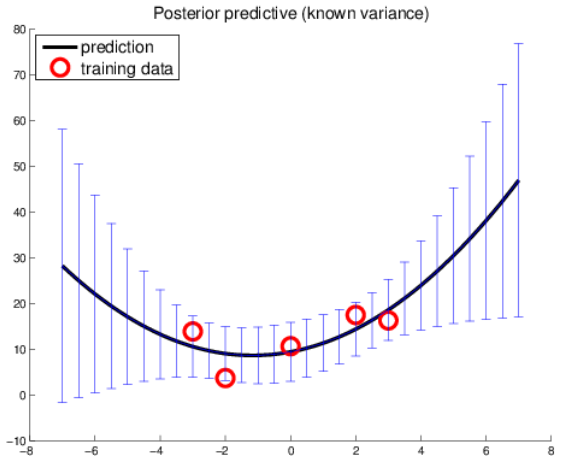
After 1 observation

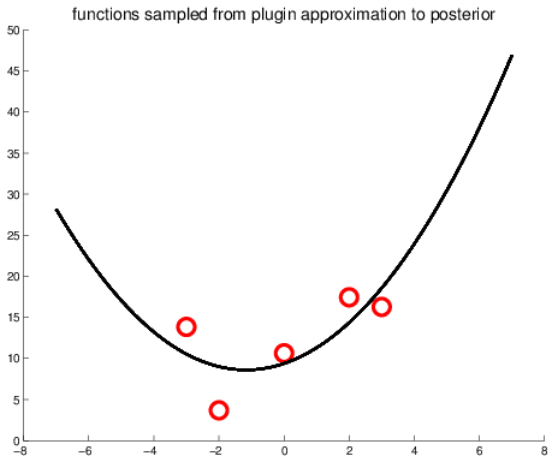
After 2 observations

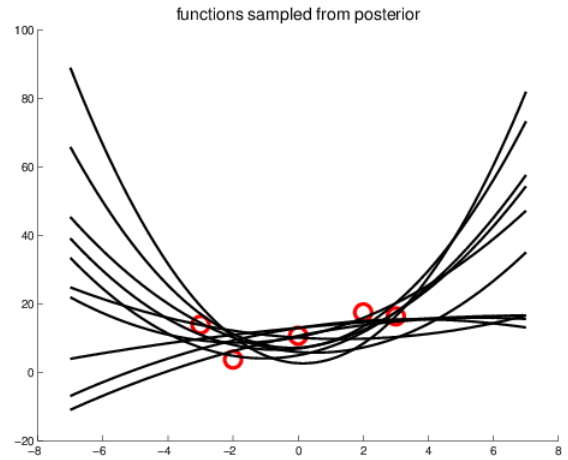
After 20 observations: we're converging on the true model (white cross)

MLE versus Posterior Predictive Confidence









The posterior predictive reflects greater uncertainty about predictions as we move outside the bounds of the observed data

Uncertainty About Model Parameters

$$\mathbf{C} = (\mathbf{X}^T\mathbf{X})^{-1}$$

$$s^2 \triangleq (\mathbf{y} - \mathbf{X}\hat{\mathbf{w}}_{mle})^T(\mathbf{y} - \mathbf{X}\hat{\mathbf{w}}_{mle})$$

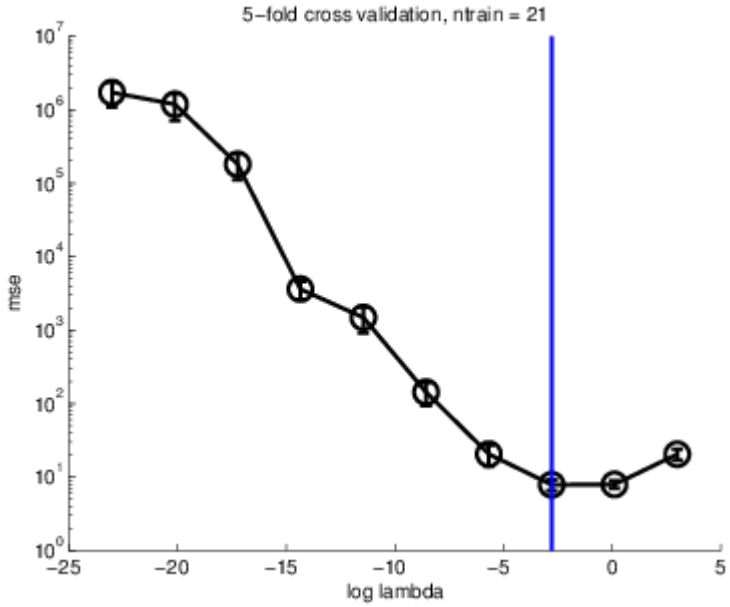
$$p(w_j|\mathcal{D}) = T(w_j|\hat{w}_j, \frac{C_{jj}s^2}{N-D}, N-D)$$

If the 95% CI of a coefficient includes zero,
that coefficient is not considered to be significant

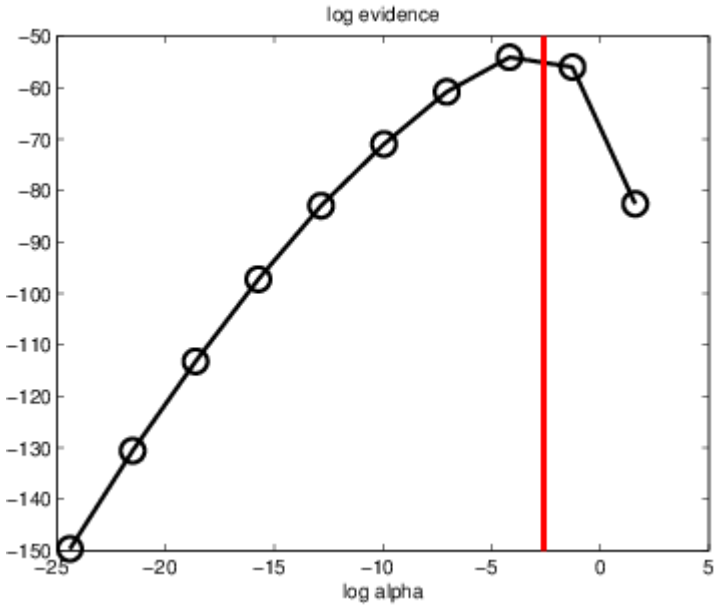
Using the caterpillar data …

w_j	$\mathbb{E}[w_j\|\mathcal{D}]$	$\sqrt{\text{var}[w_j\|\mathcal{D}]}$	95% CI	sig
w0	10.998	3.06027	[4.652, 17.345]	*
w1	-0.004	0.00156	[-0.008, -0.001]	*
w2	-0.054	0.02190	[-0.099, -0.008]	*
w3	0.068	0.09947	[-0.138, 0.274]	
w4	-1.294	0.56381	[-2.463, -0.124]	*
w5	0.232	0.10438	[0.015, 0.448]	*
w6	-0.357	1.56646	[-3.605, 2.892]	
w7	-0.237	1.00601	[-2.324, 1.849]	
w8	0.181	0.23672	[-0.310, 0.672]	
w9	-1.285	0.86485	[-3.079, 0.508]	
w10	-0.433	0.73487	[-1.957, 1.091]	

5-Fold Cross Validation versus Empirical Bayes



Repeating the cross validation result from earlier



Comparing to empirical Bayes where we are manipulating the precision of the prior

Recommendation: avoid using the training data to evaluate fit

Semi-Supervised Learning

Agenda

- Definitions
- Semi-Supervised Learning Assumptions
- Self-Training
- Co-Training
- Label Propagation
- Induction versus Transduction

Definitions

- Supervised learning: labeled data is used to construct a model
- Unsupervised learning: unlabeled data is used to construct a model
- Semi-Supervised learning: both labeled and unlabeled data are used to construct a model
  - It can be cheap to collect unlabeled data, but obtaining labels can be both expensive and time-consuming

Semi-Supervised Learning Assumptions

For classification …

- Smoothness: a decision boundary runs through a low density area
- Clustering: observations that belong to the same cluster will have the same label
- Manifold: observations can be effectively projected to a much lower dimension

Self-Training

- Use the labeled data to construct a model
- Generate predictions for the unlabeled data
- Use high-confidence predictions as labels, add those observations to the training data, and construct a new model
- Be careful!

A naïve approach may do the wrong thing

Recommendation: use repeated cross validation for evaluation

Co-Training

- Use the labeled data to construct a pair of models
  - Construct model1 using featureSet1 (e.g. images)
  - Construct model2 using featureSet2 (e.g. text descriptions)
- Generate predictions for the unlabeled data
- Use high-confidence predictions for a model as labels, add those observations to the training data for **the other** model, and construct new models
  - Add high-confidence predictions for model1 to the training set for model2
  - Add high-confidence predictions for model2 to the training set for model1

Label Propagation

1. Form the affinity matrix W defined by $W_{ij} = \exp(-\|x_i - x_j\|^2/2\sigma^2)$ if $i \neq j$ and $W_{ii} = 0$.
2. Construct the matrix $S = D^{-1/2}WD^{-1/2}$ in which D is a diagonal matrix with its (i,i)-element equal to the sum of the i-th row of W.
3. Iterate $F(t+1) = \alpha SF(t) + (1-\alpha)Y$ until convergence, where α is a parameter in $(0,1)$.
4. Let F^* denote the limit of the sequence $\{F(t)\}$. Label each point x_i as a label $y_i = \arg\max_{j \leq c} F^*_{ij}$.

http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.115.3219

The ***S*** similarity matrix is the same matrix we used for spectral decomposition for spectral clustering. This algorithm incrementally propagates labels to "neighbors."

Induction versus Transduction

Induction produces a model that can be used to make predictions for unseen data