Lecture 4: Regularization; gradient descent

TTIC 31020: Introduction to Machine Learning

Instructor: Greg Shakhnarovich

TTI-Chicago

October 6, 2016

Administrivia

- Problem set 1: out today, due in two weeks
- Problem set 2 will be out before PS1 is due
- Tutorial: next Monday 3pm (Python), in this room
- Will announce if/when will have additional tutorial slot
- I am traveling next week; lectures held as usual

Review: noise model and log-likelihood

- Statistical model: noise as a Gaussian random variable

$$y = f(\mathbf{x}; \mathbf{w}) + \nu, \qquad \nu \sim \mathcal{N}\left(\nu;\, 0, \sigma^2\right)$$

  equivalent to $p(y|\mathbf{x}; \mathbf{w}, \sigma) = \mathcal{N}\left(y;\, f(\mathbf{x}; \mathbf{w}), \sigma^2\right)$

- Maximizing log-likelihood under this model

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmax}} \sum_i \log p(y_i|\mathbf{x}_i; \mathbf{w}, \sigma)$$

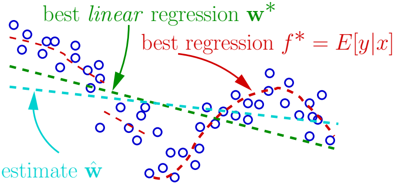
  is equivalent to least-squares regression

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_i \left(y_i - f(\mathbf{x}_i; \mathbf{w})\right)^2$$

Review: Decomposition of error

- Approximation error

$$E\left[\left(y - \mathbf{w}^{*T}\mathbf{x}\right)^2\right]$$



- Estimation error

$$E\left[\left(\mathbf{w}^{*T}\mathbf{x} - \hat{\mathbf{w}}^T\mathbf{x}\right)^2\right]$$

- Approximation error: due to the failure to include optimal predictor in the model class, plus inherent uncertainty in $y|\mathbf{x}$
- Estimation error: due to failure to select the best predictor in the chosen model class; could be reduced with more data

Review: generalized linear regression

$$f(\mathbf{x};\mathbf{w}) \;=\; w_0 + w_1\phi_1(\mathbf{x}) + w_2\phi_2(\mathbf{x}) + \ldots + w_m\phi_m(\mathbf{x}),$$

- Still the same ML estimation technique applies:

$$\hat{\mathbf{w}} \;=\; \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$$

where $\mathbf{X}$ is the *design matrix*

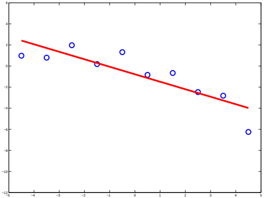
$$\begin{bmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \ldots & \phi_m(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \ldots & \phi_m(\mathbf{x}_2) \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \phi_2(\mathbf{x}_N) & \ldots & \phi_m(\mathbf{x}_N) \end{bmatrix}$$

Roadmap

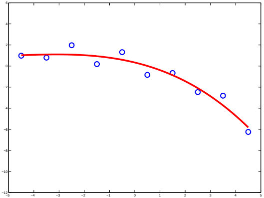
- So far: least squares regression (with arbitrary feature functions)
  - Closed form solution for maximum likelihood
  - Overfitting is a problem
- Today: regularization – main tool to combat overfitting
- Also: gradient descent as an alternative to closed form solution

Model complexity and overfitting

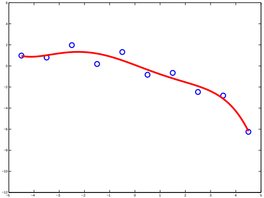
- Data drawn from 3rd order model:



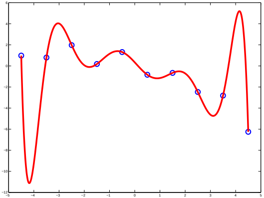
$m = 1$



$m = 3$

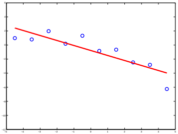


$m = 5$

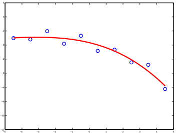


$m = 10$

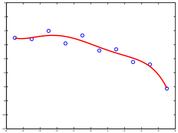
Controlling for overfitting



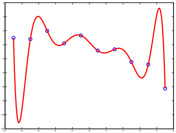
$L = 1.4, \hat{L}_{\mathsf{cv}} = 2.6$



$L = 0.4, \hat{L}_{\mathsf{cv}} = 1.3$



$L = 0.3, \hat{L}_{\mathsf{cv}} = 2.7$

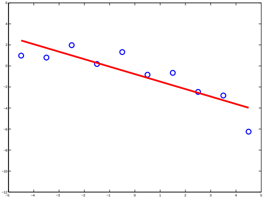


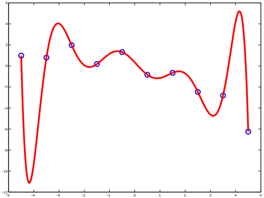
$L = 0, \hat{L}_{\mathsf{cv}} = 4 \times 10^4$

- More complex model (10th degree) overfits more than simple model (linear)
- Pure ERM would always prefer complex models
- Holdout/validation/cross-validation is a way to control for this in *model selection*

Model complexity - intuition

- Intuitively, the complexity of the model can be measured by the number of "degrees of freedom" (independent parameters).
  - The more complex the model, the more data needed to fit
    $\Rightarrow$ For a given number of points, a more complex model more likely to overfit.





$m = 1$, 2 parameters $m = 10$, 11 parameters

- This is an issue only because of finite training data!

Penalizing model complexity

- Idea 1: restrict model complexity based on amount of data
  - Rule of thumb: ≈ 10 examples per parameter

Penalizing model complexity

- Idea 1: restrict model complexity based on amount of data
  - Rule of thumb: $\approx$ 10 examples per parameter
- Idea 2: directly penalize by the number of parameters.
  Akaike information criterion (AIC): maximize

$$\log p\left(X \mid \widehat{\mathbf{w}}\right) - \#\text{params}$$

Penalizing model complexity

- Idea 1: restrict model complexity based on amount of data
  - Rule of thumb: $\approx$ 10 examples per parameter
- Idea 2: directly penalize by the number of parameters.
  Akaike information criterion (AIC): maximize

$$\log p\left(X \,|\widehat{\mathbf{w}}\right) - \#\text{params}$$

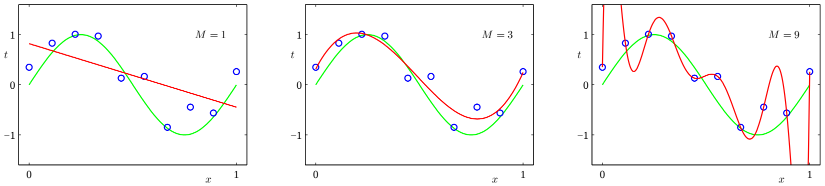
- But: Definition of model complexity as a number of parameters is a bit too simplistic. Consider feature vector

$$\boldsymbol{\phi} x = \begin{bmatrix} 1 & x & -2x & 2x & x^2 & \frac{1}{2}x^2 \end{bmatrix}$$

  Does linear regression $\boldsymbol{\phi}(x) \to y$ really have 6 parameters?
- Idea: look at the behavior of the values of $\mathbf{w}^*$

Linear regression complexity

- Example: polynomial regression, true [from Bishop, Ch. 1]



- Value of the optimal (ML) regression coefficients:

	$m=0$	$m=1$	$m=3$	$m=9$
w_0^*	0.19	0.82	0.31	0.35
w_1^*		-1.27	7.99	232.37
w_2^*			-25.43	-5321.83
w_3^*			17.37	48568.31
w_4^*				-231639.30
w_5^*				640042.26
w_6^*				-1061800.52
w_7^*				1042400.18
w_8^*				-557682.99
w_9^*				125201.43

Description length

- Intuition: should penalize not the parameters, but the number of bits required to encode the parameters
- With finite set of parameter values, these are equivalent
- With "infinite" set, we can limit the effective number of degrees of freedom by restricting the value of the parameters.
- Then we have penalized log-likelihood:

$$\mathbf{w}^* = \operatorname*{argmax}_{\mathbf{w}} \left\{ \frac{1}{2} \sum_{i=1}^{N} \log p(\text{data}_i;\, \mathbf{w}) - \text{penalty}(\mathbf{w}) \right\}$$

Shrinkage methods

- Shrinkage methods impose penalty on the size of $\mathbf{w}$
- We can measure "size" in a few different ways. Let us start with L_2 norm:

$$\mathbf{w}^*_{\text{ridge}} = \operatorname*{argmax}_{\mathbf{w}} \left\{ \sum_{i=1}^{N} \log p(\text{data}_i;\, \mathbf{w}) - \lambda \|\mathbf{w}\|^2 \right\}$$

  in regression "data_i" $= y_i | \mathbf{x}_i$
- This is *ridge regression*; λ is the *regularization* parameter
- Does it matter that log-likelihood is not averaged?

Shrinkage methods

- Shrinkage methods impose penalty on the size of $\mathbf{w}$
- We can measure "size" in a few different ways. Let us start with L_2 norm:

$$\mathbf{w}^*_{\text{ridge}} = \operatorname*{argmax}_{\mathbf{w}} \left\{ \sum_{i=1}^{N} \log p(\text{data}_i;\, \mathbf{w}) - \lambda \|\mathbf{w}\|^2 \right\}$$

in regression "data$_i$" $= y_i | \mathbf{x}_i$

- This is *ridge regression*; λ is the *regularization* parameter
- Does it matter that log-likelihood is not averaged? Consider relative effect of the value of λ

Ridge regression

$$\mathbf{w}^*_{\text{ridge}} = \underset{\mathbf{w}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2 + \lambda \sum_{j=1}^{m} w_j^2 \right\}$$

- Recall: $\mathbf{w} = [w_0, w_1, \ldots, w_m]$
- Usually do not include w_0 in regularization (why?)

Ridge regression

$$\mathbf{w}^*_{\text{ridge}} = \operatorname*{argmin}_{\mathbf{w}} \left\{ \sum_{i=1}^{N} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2 + \lambda \sum_{j=1}^{m} w_j^2 \right\}$$

- Recall: $\mathbf{w} = [w_0, w_1, \ldots, w_m]$
- Usually do not include w_0 in regularization (why?)
- Closed form solution:

$$\widehat{\mathbf{w}}^*_{\text{ridge}} = \left(\lambda \mathbf{I} + \mathbf{X}^T \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{y}.$$

- Careful: solution *not* invariant to scaling! Should normalize input before solving.

Lasso regression

- The L_1-penalized maximum likelihood under Gaussian noise model:

$$\mathbf{w}^*_{\text{lasso}} = \underset{\mathbf{w}}{\operatorname{argmax}} \left\{ -\sum_{i=1}^{N} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2 - \lambda \sum_{j=1}^{m} |w_j| \right\}$$

- This is still concave (i.e. unique maximum), but not “smooth” (differentiable).
- Can solve it efficiently using convex programming methods or first-order numerical optimization (gradient descent)
- Why is it called “lasso”?

Lasso regression

- The L_1-penalized maximum likelihood under Gaussian noise model:

$$\mathbf{w}^*_{\text{lasso}} = \operatorname*{argmax}_{\mathbf{w}} \left\{ -\sum_{i=1}^{N} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2 - \lambda \sum_{j=1}^{m} |w_j| \right\}$$

- This is still concave (i.e. unique maximum), but not “smooth” (differentiable).
- Can solve it efficiently using convex programming methods or first-order numerical optimization (gradient descent)
- Why is it called “lasso”?
  least absolute shrinkage and selection operator

Optimization of ridge regression

- Can rewrite the optimization problem

$$\min_{\mathbf{w}} \sum_{i=1}^{N} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2 + \lambda \sum_{j=1}^{m} w_j^2$$

in the proper objective/constraint form:

$$\min_{\mathbf{w}} \sum_{i=1}^{N} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2$$

$$\text{subject to} \sum_{j=1}^{m} w_j^2 \leq t$$

- Correspondence $\lambda \Rightarrow t$ can be shown using Lagrange multipliers.

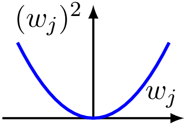
Optimization for Lasso

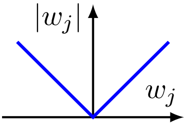
- Similarly, for Lasso:

$$\min_{\mathbf{w}} \sum_{i=1}^{N} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2$$

$$\text{subject to} \sum_{j=1}^{m} |w_j| \leq t$$

- Compare shape of the penalty as a function of w_j:

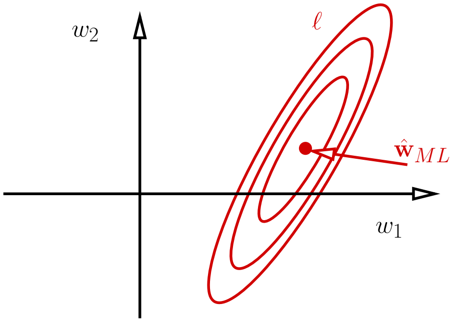




Lasso vs. ridge: geometry of error surfaces

- An equivalent formulation for L_p regularization: constrained maximization

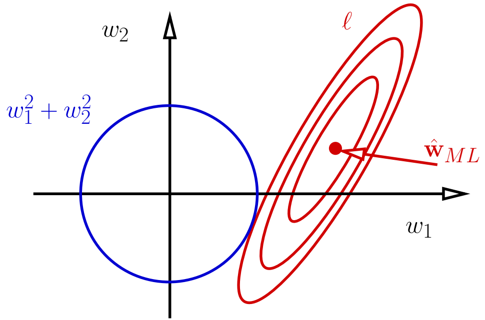
$$\hat{\mathbf{w}} = \underset{\mathbf{w}:\, \sum_{j=1}^{m} |w_j|^p \leq \beta}{\operatorname{argmax}} - \sum_{i=1}^{N} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2.$$



Lasso vs. ridge: geometry of error surfaces

- An equivalent formulation for L_p regularization: constrained maximization

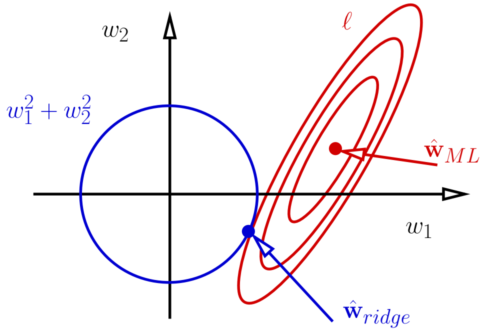
$$\hat{\mathbf{w}} = \underset{\mathbf{w}:\, \sum_{j=1}^{m} |w_j|^p \leq \beta}{\operatorname{argmax}} \; -\sum_{i=1}^{N} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2.$$



Lasso vs. ridge: geometry of error surfaces

- An equivalent formulation for L_p regularization: constrained maximization

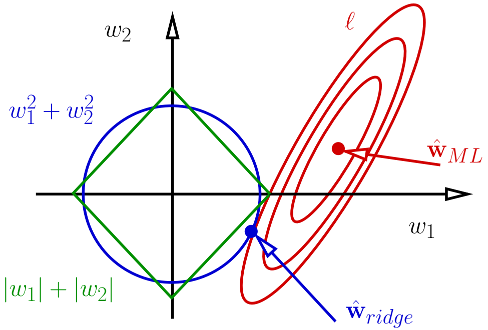
$$\hat{\mathbf{w}} = \underset{\mathbf{w}:\, \sum_{j=1}^{m} |w_j|^p \le \beta}{\operatorname{argmax}} -\sum_{i=1}^{N} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2.$$



Lasso vs. ridge: geometry of error surfaces

- An equivalent formulation for L_p regularization: constrained maximization

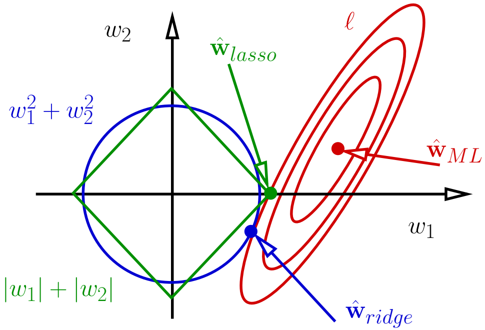
$$\hat{\mathbf{w}} = \operatorname*{argmax}_{\mathbf{w}:\, \sum_{j=1}^{m} |w_j|^p \le \beta} - \sum_{i=1}^{N} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2.$$



Lasso vs. ridge: geometry of error surfaces

- An equivalent formulation for L_p regularization: constrained maximization

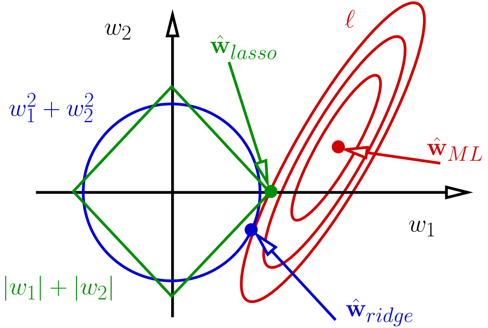
$$\hat{\mathbf{w}} = \underset{\mathbf{w}:\, \sum_{j=1}^{m} |w_j|^p \leq \beta}{\operatorname{argmax}} \; -\sum_{i=1}^{N} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2.$$



Lasso vs. ridge: geometry of error surfaces

- An equivalent formulation for L_p regularization: constrained maximization

$$\hat{\mathbf{w}} \;=\; \underset{\mathbf{w}:\,\sum_{j=1}^{m}|w_j|^p\le\beta}{\operatorname{argmax}} \; -\sum_{i=1}^{N}(y_i-\mathbf{w}\cdot\mathbf{x}_i)^2.$$

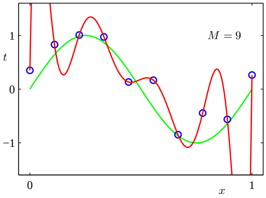


- With sufficiently large λ (=sufficiently small β) lasso leads to *sparsity*.
- Must explicitly solve the above optimization problem – e.g., using Lagrange multipliers.

Choice of λ

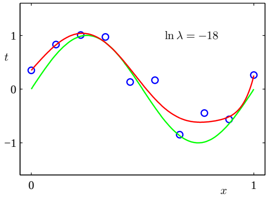
- Example [from Bishop, Ch. 1]: 9th deg polynomial with varying λ:

$\lambda = 0$



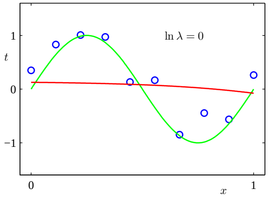
$\|\mathbf{w}^*\|^2 > 10^{12}$

$\lambda = e^{-18}$



$\|\mathbf{w}^*\|^2 \approx 21595$

$\lambda = 1$

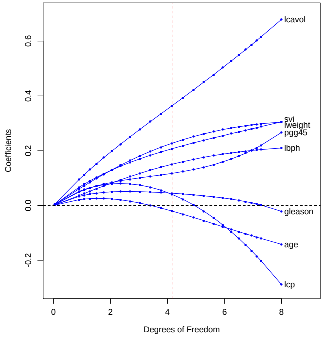


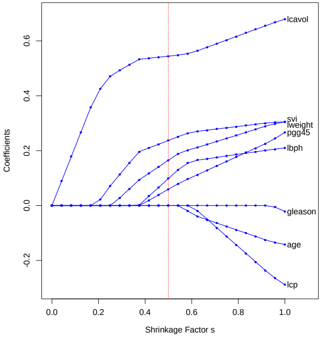
$\|\mathbf{w}^*\|^2 \approx 0.027$

- Most often: choose λ by (cross) validation

Example: lasso vs. ridge regularization paths

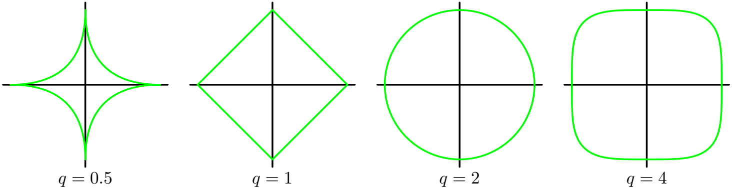
- Example: prostate data [Hastie, Tibshirani and Friedman]
  Red lines: choice of λ by 10-fold CV.





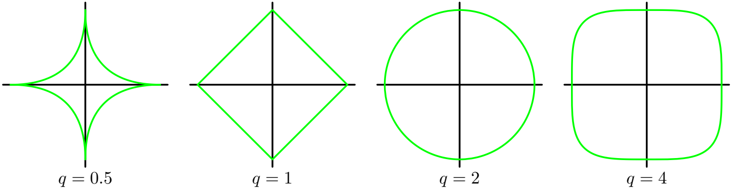
General view of L_q penalty

- Can be creative in design of penalty function $\|\mathbf{w}\|_q$



General view of L_q penalty

- Can be creative in design of penalty function $\|\mathbf{w}\|_q$



- For $q > 1$, no sparsity is achieved.
- For $q < 1$, non-convex
- What about L_0?

$$\min_{\mathbf{w}} \sum (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2 \quad \text{s.t. } |\{w_j : w_j > 0\}| \leq M$$

is NP-hard