MACHINE LEARNING

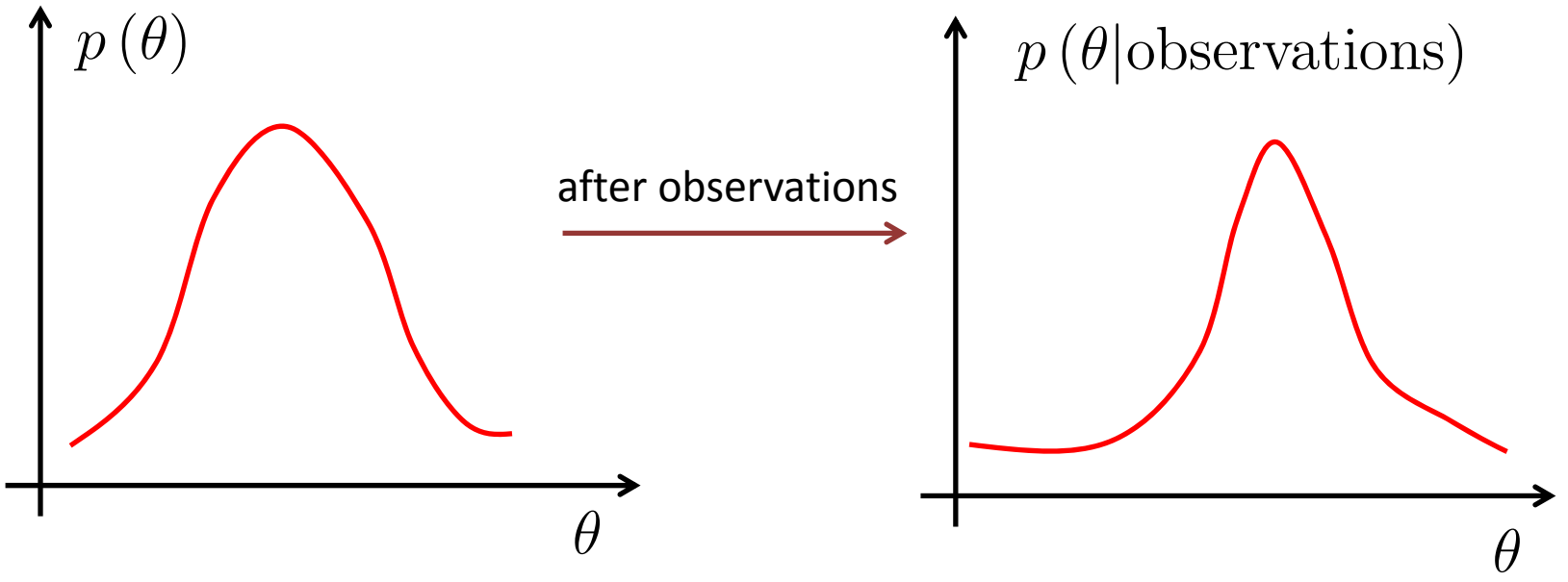
CHAPTER 2: LINEAR MODELS FOR REGRESSION

LINEAR MODELS FOR REGRESSION

1. The concept of regression
2. Maximum Likelihood and Least Square
3. Over-fitting and Regularization
4. The Bias-Variance Trade-off
5. Bayesian Linear Regression
6. Sparse regression

What about prior

Rather than estimating a single θ, we obtain a distribution over possible values of θ



Bayesian Learning

Use Bayes rule:

posterior probability: $p(\theta|\mathcal{D})$

Likelihood: $p(\mathcal{D}|\theta)$

prior: $p(\theta)$

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)\,p(\theta)}{p(\mathcal{D})}$$

Or equivalently:

$$p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)\,p(\theta)$$

A likelihood function is a function of the parameters.

Bayesian Learning

Bayes rule:

$$p\left(\theta|\mathcal{D}\right) \propto p\left(\mathcal{D}|\theta\right) p\left(\theta\right)$$

What about prior?

Represent expert knowledge

Learning a Gaussian

Collect a bunch of data

- Hopefully, i.i.d. samples
- e.g. exam scores

English exam score

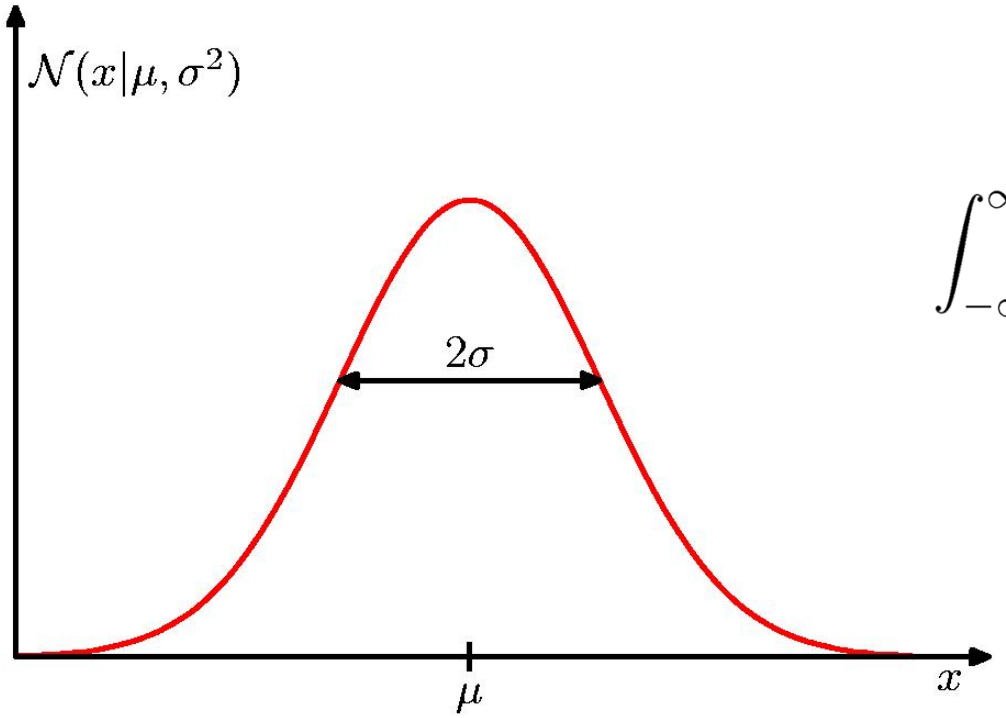
$$x_1 = 60,$$
$$x_2 = 30,$$
$$x_3 = 70,$$
$$x_4 = 80,$$
$$x_5 = 90$$

Learn parameters

- Mean
- Variance

The Gaussian Distribution

$$\mathcal{N}\left(x|\mu,\sigma^2\right) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$



$$\mathcal{N}(x|\mu,\sigma^2) > 0$$

$$\int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right)\,\mathrm{d}x = 1$$

Learning a Gaussian

We suppose English exam score satisfy Gaussain Distribution

Prob. of i.i.d. samples $\mathcal{D} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$

$$p\left(\mathcal{D}|\mu,\sigma\right) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^N \prod_n exp\left\{-\frac{1}{2\sigma^2}\left(x_n-\mu\right)^2\right\}$$

$$\mu_{\mathrm{ML}}, \sigma_{\mathrm{ML}} = \mathrm{argmax}\ p\left(\mathcal{D}|\mu,\sigma\right)$$

Learning a Gaussian

Log-likelihood of data:

$$\ln\ p\left(\mathcal{D}|\mu,\sigma\right) = \ln\left(\frac{1}{\sigma\sqrt{2\pi}}\right)^{N}\prod_{n} exp\left\{-\frac{1}{2\sigma^{2}}\left(x_{n}-\mu\right)^{2}\right\}$$

$$= -N\ln\sigma\sqrt{2\pi} - \sum_{n=1}^{N}\frac{1}{2\sigma^{2}}\left(x_{n}-\mu\right)^{2}$$

Learning a Gaussian

What's MLE for mean?

$$\frac{d}{d\mu} \ln p\left(\mathcal{D}|\mu, \sigma\right) = 0$$

$$\mu_{\mathrm{ML}} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

Learning a Gaussian

Again, set derivative to zero:

$$\frac{d}{d\sigma} \ln p\left(\mathcal{D}|\mu, \sigma\right) = 0$$

$$\sigma^2_{\mathrm{ML}} = \frac{1}{N} \sum_{i=1}^{N} \left(x_i - \mu_{\mathrm{ML}}\right)^2$$

Expected result of estimation is not true parameter!
biased variance estimation!

Learning a Gaussian

$$\mu_{\mathrm{ML}} = \frac{1}{5}\sum_{i=1}^{5}(60 + 30 + 70 + 80 + 90) = 66$$

$$\sigma^2_{\mathrm{ML}} = \frac{1}{5}\sum_{i=1}^{5}(x_i - \mu_{\mathrm{ML}})^2 = 424$$

English exam score

$x_1 = 60,$
$x_2 = 30,$
$x_3 = 70,$
$x_4 = 80,$
$x_5 = 90$

Bayesian learning for Gaussian parameters

Conjugate priors

Mean: Gaussian prior

Variance: Wishart Distribution

- Prior for mean

$$p\left(\mu|\eta,\lambda\right)=\frac{1}{\lambda\sqrt{2\pi}}exp\left\{-\frac{1}{2\lambda^{2}}\left(\mu-\eta\right)^{2}\right\}$$

MAP: Maximum a posterior approximation

Bayes rule:

$$p\left(\mu|\mathcal{D}\right) \propto p\left(\mathcal{D}|\mu\right) p\left(\mu\right)$$

$$p\left(\mu|\eta,\lambda\right) = \frac{1}{\lambda\sqrt{2\pi}} exp\left\{-\frac{1}{2\lambda^2}\left(\mu-\eta\right)^2\right\}$$

$$p\left(\mathcal{D}|\mu,\sigma\right) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right) \prod_i exp\left\{-\frac{1}{2\sigma^2}\left(x_i-\mu\right)^2\right\}$$

MAP: Maximum a posterior approximation

$$\frac{d}{d\mu}\left[\ln p\left(\mathcal{D}|\mu\right) p\left(\mu\right)\right] = \frac{d}{d\mu}\left[\ln p\left(\mathcal{D}|\mu\right) + \ln p\left(\mu\right)\right]$$

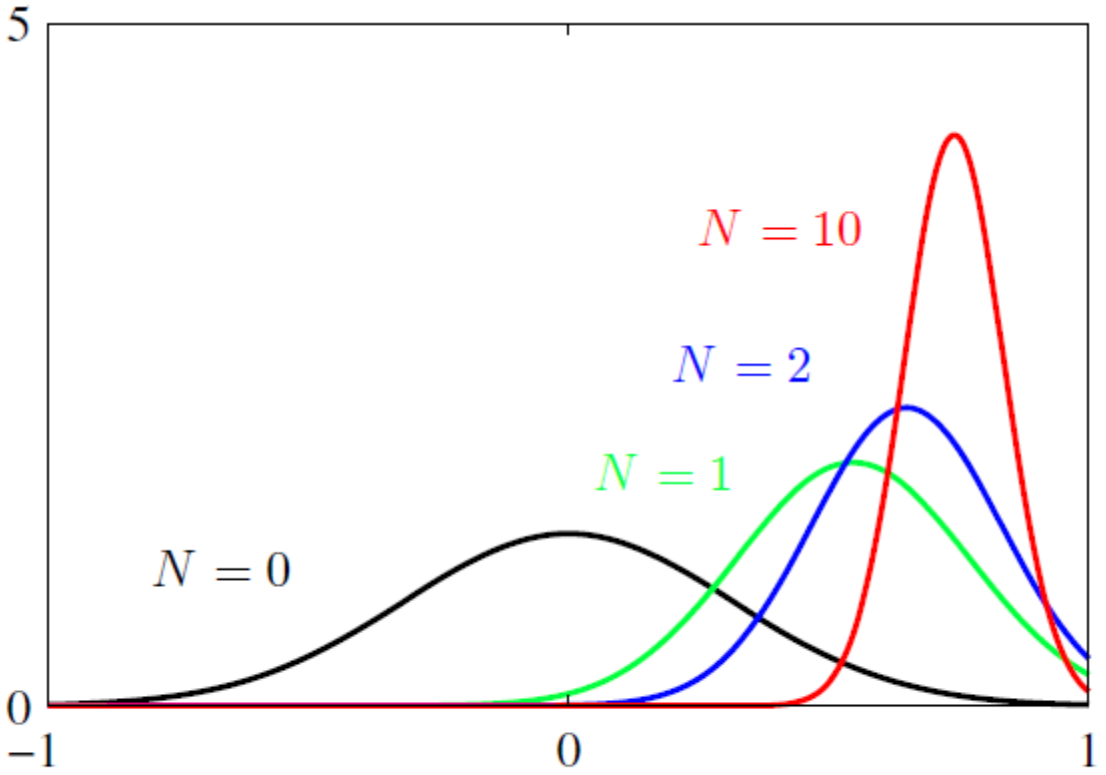
posterior likelihood prior

set the above equation to zero

$$\mu_{\mathrm{MAP}} = \frac{\sigma^2}{\sigma^2 + N\lambda^2}\mu_{\mathrm{ML}} + \frac{\sigma^2}{\sigma^2 + N\lambda^2}\eta$$

MAP: Maximum a posterior approximation

The data points are generated from a Gaussian of mean 0.8 and variance 0.1, and the prior is chosen to have mean 0. In both the prior and the likelihood function, the variance is set to the true values.



Posterior distribution

MAP: Maximum a posterior approximation

5 observations are very insufficient.

MAP estimation for the mean of exam score:

The mean of exam score has the below prior:

$$p(\mu) \sim \mathcal{N}(\mu|70, 25)$$

English exam score

$$x_1 = 60,$$
$$x_2 = 30,$$
$$x_3 = 70,$$
$$x_4 = 80,$$
$$x_5 = 90$$

MAP: Maximum a posterior approximation

Maximum likelihood estimation:

$$\mu_{\mathrm{ML}} = \frac{1}{N}\sum_{i=1}^{N} x_i = 66$$

$$\mu_{\mathrm{MAP}} = \frac{\sigma^2}{\sigma^2 + N\lambda^2}\mu_{\mathrm{ML}} + \frac{\sigma^2}{\sigma^2 + N\lambda^2}\eta = 69.1$$

where $\sigma^2 = 424$, $\lambda^2 = 25$ and $\eta = 70$, which are known.

MAP: Maximum a posterior approximation

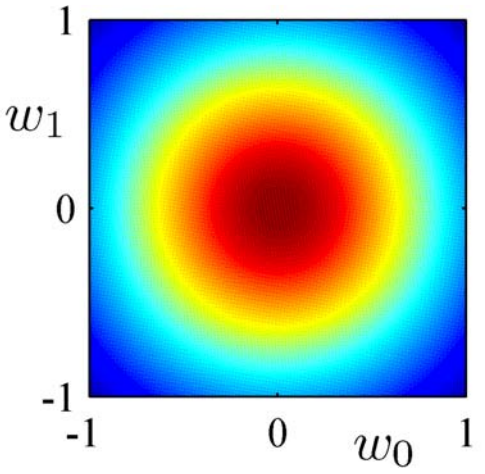
From the result, we know the MAP estimation use the observations of variables and the priority of parameters. Therefore it is the combination of maximum likelihood estimation and parameter priority.

When the observations are very limited, MAP estimation is much more robust than MLE estimation.

Bayesian Linear Regression

Define a conjugate prior over $\mathbf{w}$

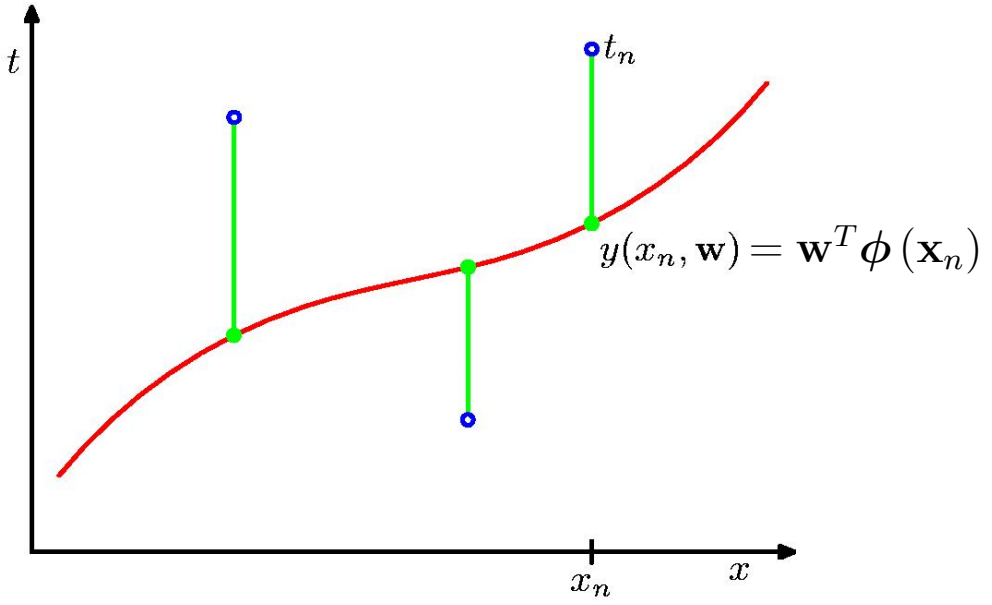
$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I})$$



Bayesian Linear Regression

The likelihood function of $\mathbf{w}$ when given one observation pairs $\{\mathbf{x}_n, t_n\}$.

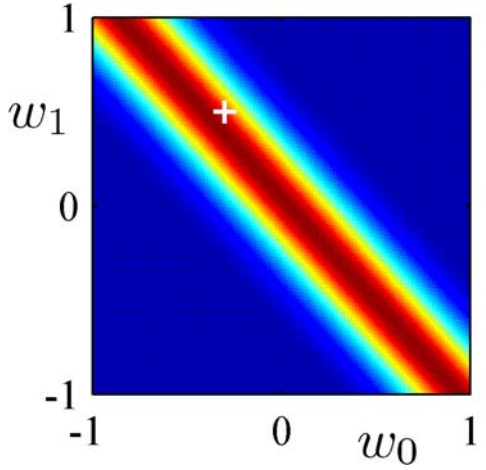
$$p\left(t_n|\mathbf{x}_n, \mathbf{w}\right) = \mathcal{N}\left(t_n|\mathbf{w}^T\boldsymbol{\phi}\left(\mathbf{x}_n\right), \beta^{-1}\right)$$

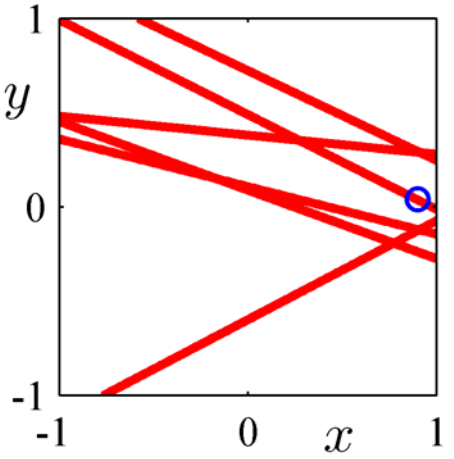


Bayesian Linear Regression

The likelihood function of $\mathbf{w}$ when given one observation pairs $\{\mathbf{x}_n, t_n\}$.

$$p\left(t_n|\mathbf{x}_n, \mathbf{w}\right) = \mathcal{N}\left(t_n|\mathbf{w}^T\boldsymbol{\phi}\left(\mathbf{x}_n\right), \beta^{-1}\right)$$





Bayesian Linear Regression

The likelihood function of $\mathbf{w}$ when given all the observation pairs $\{\mathbf{x}_n, t_n\}_{n=1}^{N}$.

$$p\left(\mathbf{t}|\mathbf{X}, \mathbf{w}\right) = \prod_{n=1}^{N} \mathcal{N}\left(t_n|\mathbf{w}^T\boldsymbol{\phi}\left(\mathbf{x}_n\right), \beta^{-1}\right)$$

Bayesian Linear Regression

Combining this with the likelihood function and using results for marginal and conditional Gaussian distributions, gives the posterior

$$p(\mathbf{w}|\mathbf{t}, \mathbf{X}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$$

where

$$\begin{aligned} \mathbf{m}_N &= \beta \mathbf{S}_N \mathbf{\Phi}^{\mathrm{T}} \mathbf{t} \\ \mathbf{S}_N^{-1} &= \alpha \mathbf{I} + \beta \mathbf{\Phi}^{\mathrm{T}} \mathbf{\Phi}. \end{aligned}$$

Bayesian Linear Regression

The log of the posterior distribution is given by

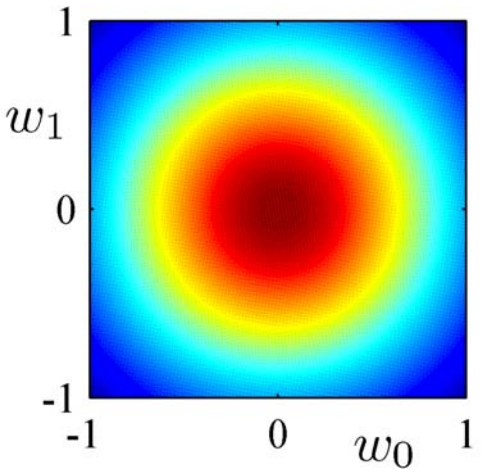
$$\ln p\left(\mathbf{w}|\mathbf{t}, \mathbf{X}\right)= -\frac{\beta}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 - \frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \text{const.}$$

Next we consider an example …

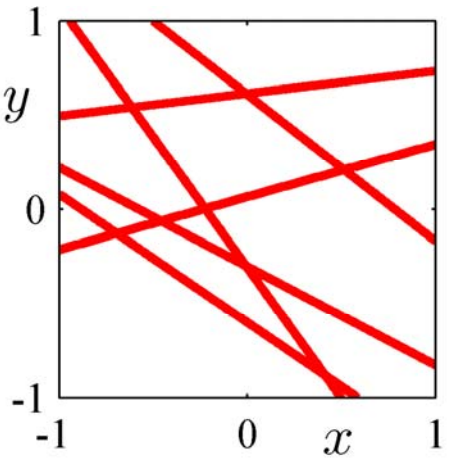
Bayesian Linear Regression

0 data points observed

Prior



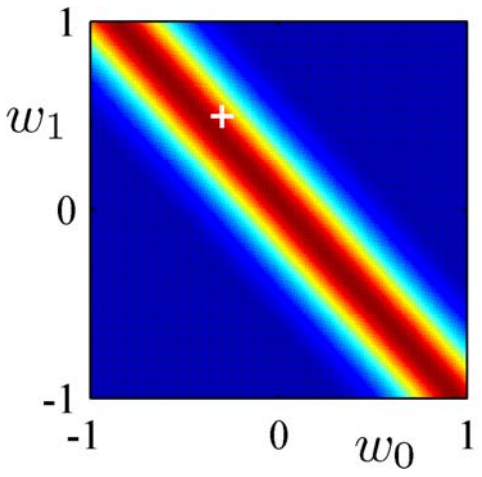
Data Space



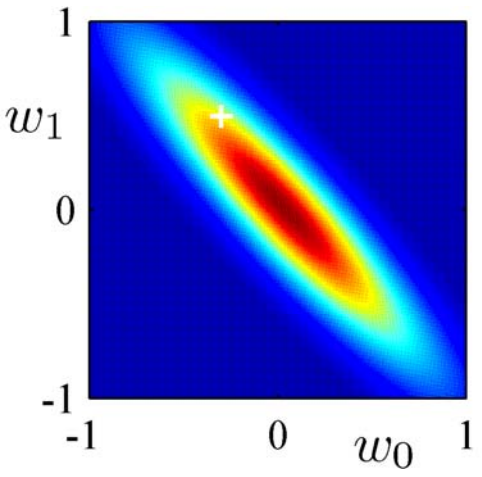
Bayesian Linear Regression

1 data point observed

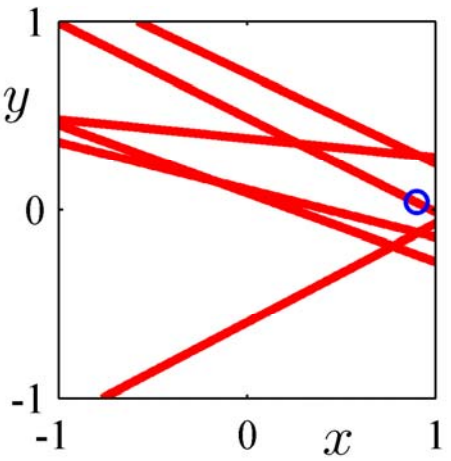
Likelihood



Posterior



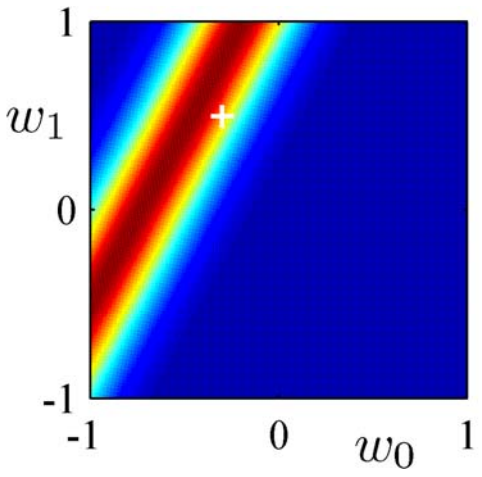
Data Space



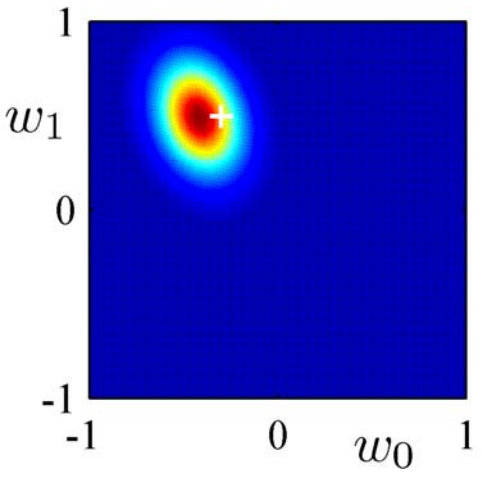
Bayesian Linear Regression

2 data points observed

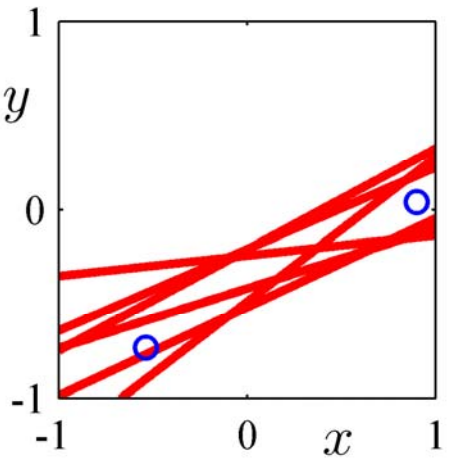
Likelihood



Posterior



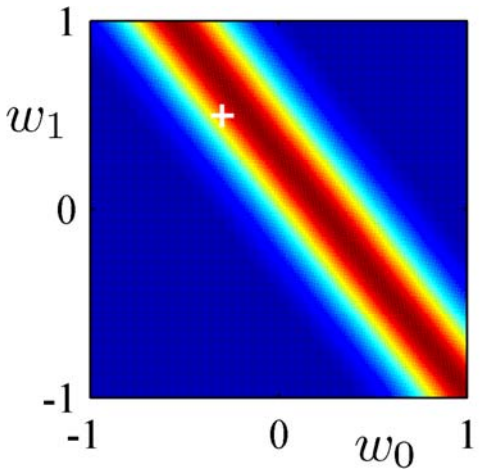
Data Space



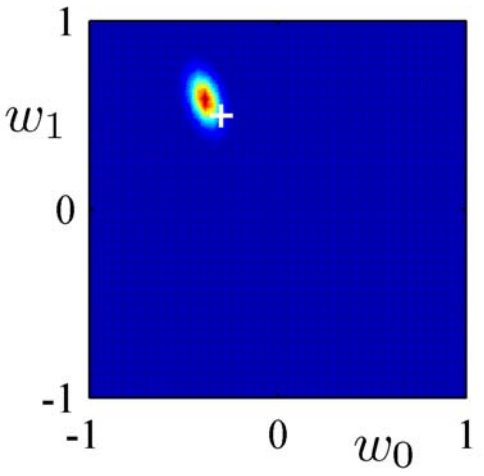
Bayesian Linear Regression

20 data points observed

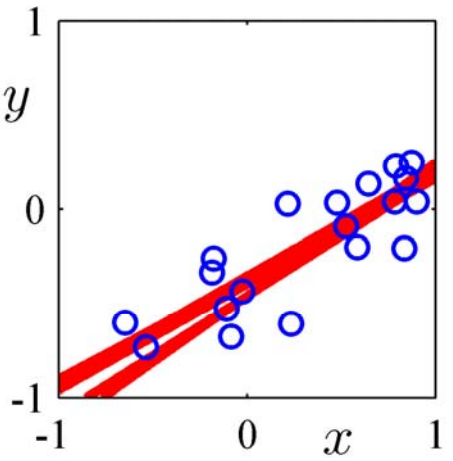
Likelihood



Posterior



Data Space



LINEAR MODELS FOR REGRESSION

1. The concept of regression
2. Maximum Likelihood and Least Square
3. Over-fitting and Regularization
4. The Bias-Variance Trade-off
5. Bayesian Linear Regression
6. Sparse regression

Sparse regression

Vector **w** is sparse, if many entries are zero:

Very useful for many tasks, e.g.,

Efficiency: If size(**w**) = 100B, each prediction is expensive:

If part of an online system, too slow

If **w** is sparse, prediction computation only depends on number of non-zeros

Interpretability: What are the relevant dimension to make a prediction?

E.g., what are the parts of the brain associated with particular words?

But computationally intractable to perform "all subsets" regression

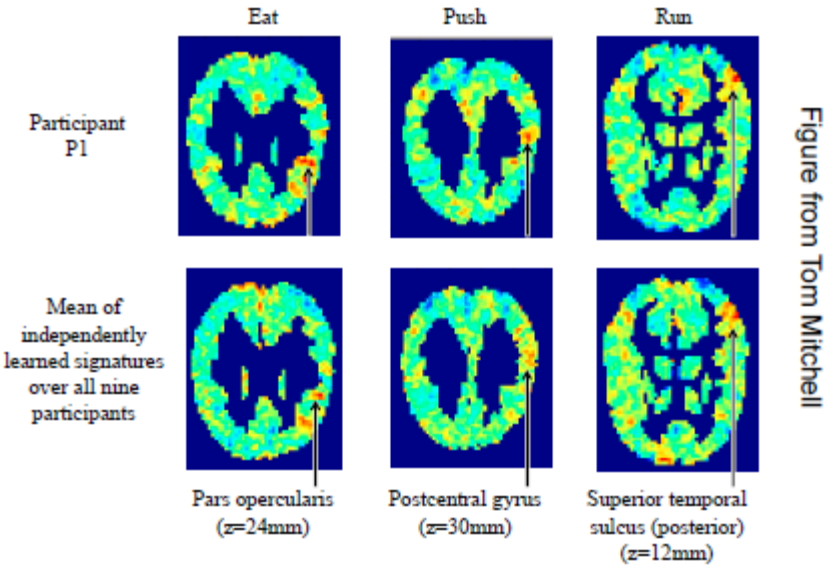


Figure from Tom Mitchell

Sparse regression example

prostate cancer data

8 features and 67 training cases

Term	LS	Best Subset	Ridge	Lasso
Intercept	2.452	2.481	2.479	2.480
lcavol	0.716	0.651	0.656	0.653
lweight	0.293	0.380	0.300	0.297
age	-0.143	-0.000	-0.129	-0.119
lbph	0.212	-0.000	0.208	0.200
svi	0.310	-0.000	0.301	0.289
lcp	-0.289	-0.000	-0.260	-0.236
gleason	-0.021	-0.000	-0.019	0.000
pgg45	0.277	0.178	0.256	0.226
Test Error	0.586	0.572	0.580	0.564

Simple greedy model selection algorithm

1. Pick a dictionary of features

   e.g., polynomials for linear regression

2. Greedy heuristic:

   Start from empty set of features $F_0 = \emptyset$

   Run learning algorithm for current set of features F_t

   obtain coefficient $\mathbf{w}_t$

   Select next best features x_i^*

   e.g. x_j That results in lowest training error learner when learning with $F_t + \{x_j\}$

   $F_{t+1} = F_t + \{x_i^*\}$

   recurse

Variable Selection by Regularization

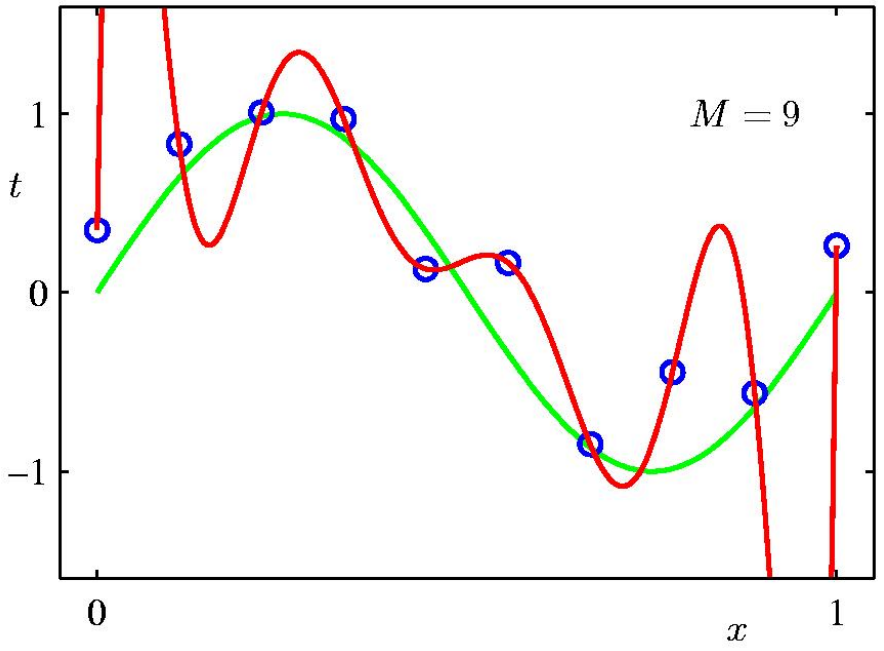
Ridge regression: Penalizes large weights

What if we want to perform “feature selection”?

E.g., Which regions of the brain are important for word prediction?

Can’t simply choose features with largest coefficients in ridge solution

Variable Selection by Regularization



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

Variable Selection by Regularization

Try new penalty: Penalize non-zero weights

Regularization penalty:

$$\|\mathbf{w}\|_1$$

Leads to sparse solutions

Just like ridge regression, solution is indexed by a continuous parameter λ

This simple approach has changed statistics, machine learning &electrical engineering

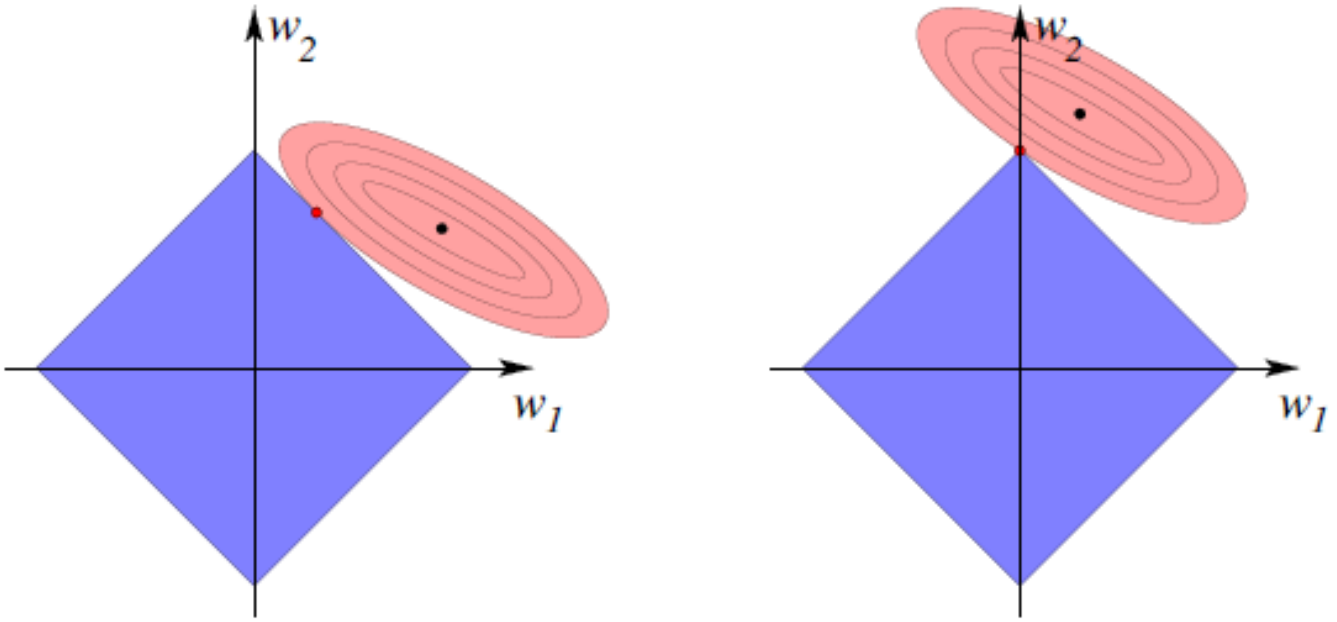
LASSO Regression

LASSO: least absolute shrinkage and selection operator

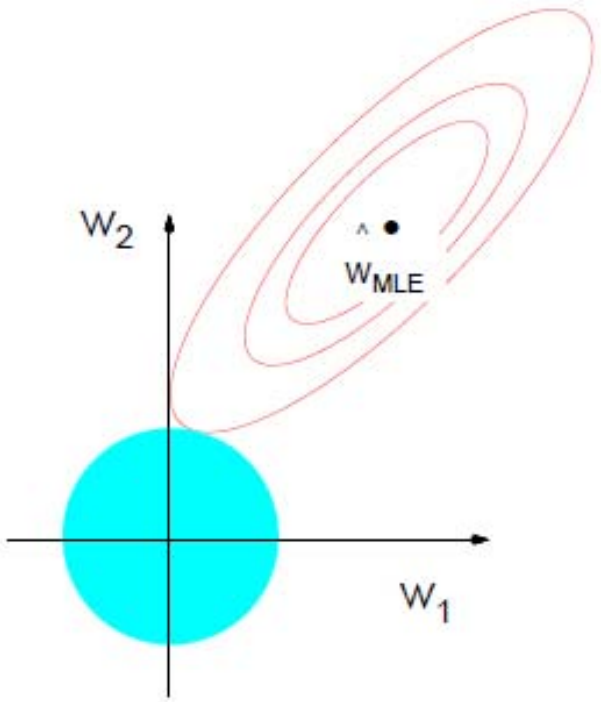
New objective:

$$\min \frac{1}{2} \sum_{n=1}^{N} \left\{ t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n) \right\}^2 + \lambda \sum_{j=1}^{M} |w_j|$$

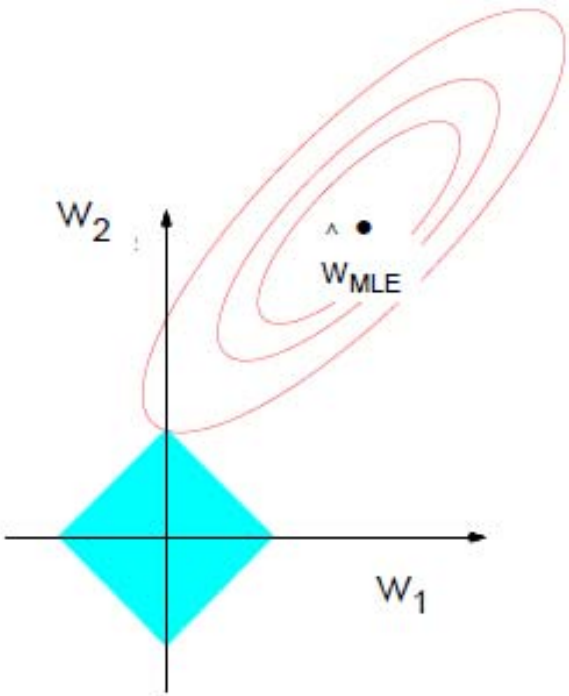
Geometric Intuition for Sparsity



Geometric Intuition for Sparsity



Ridge Regression



Lasso

Optimizing the LASSO Objective

Lasso solution:

$$\hat{\mathbf{w}} = argmin_w \frac{1}{2} \sum_{n=1}^{N} \left\{ t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n) \right\}^2 + \lambda \sum_{j=1}^{M} |w_j|$$

Take derivative and set to zero, how about the derivative of $|w_j|$?

Optimizing the LASSO Objective

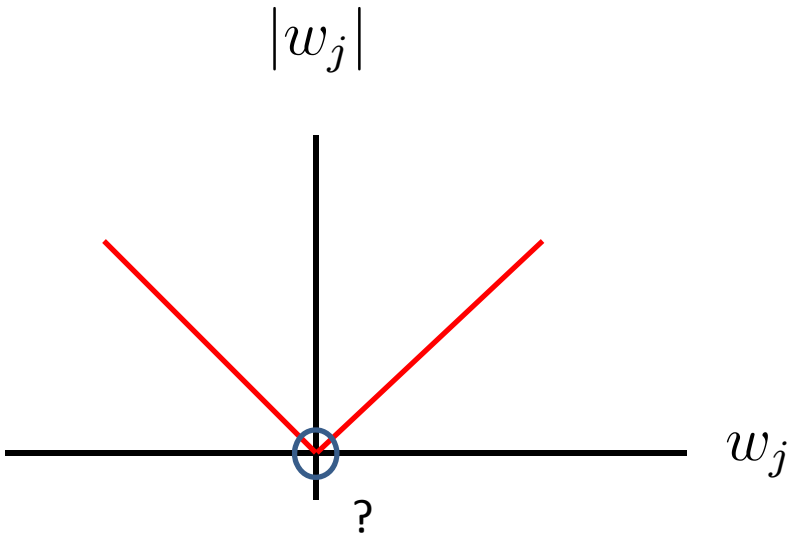
Lasso solution:

$$\hat{\mathbf{w}} = argmin_w \frac{1}{2} \sum_{n=1}^{N} \left\{ t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n) \right\}^2 + \lambda \sum_{j=1}^{M} |w_j|$$

Take the derivative and set it to be 0

1. Derivative of $|w_j|$
2. Even if you could take derivative, no closed-form solution to $\hat{\mathbf{w}}$

Optimizing the LASSO Objective



1. Derivative of $|w_j|$
2. Even if you could take derivative, no closed-form solution to $\hat{\mathbf{w}}$

Coordinate Descent

Given a function F

Want to find minimum

$$\hat{\mathbf{w}} = \text{argmin} F\left(w_0, w_1, ..., w_K\right)$$

Often, hard to find minimum for all coordinates, but easy for one coordinate.

Coordinate Descent

Initialize $\mathbf{w} = 0$ or something else

While not converged

Pick coordinate l

$$\hat{w}_l = \text{argmin} F\left(w_0, w_1, ..., w_{l-1}, w_l, w_{l+1}, ..., w_M\right)$$

How do we pick next coordinate?

random,…

Converges to optimal in LASSO

Optimizing LASSO objective function

$$\frac{1}{2}\sum_{n=1}^{N}\left\{t_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)\right\}^2 + \lambda\sum_{j=1}^{M}|w_j|$$

$RSS\,(\mathbf{w})$ reg

Taking the derivative:

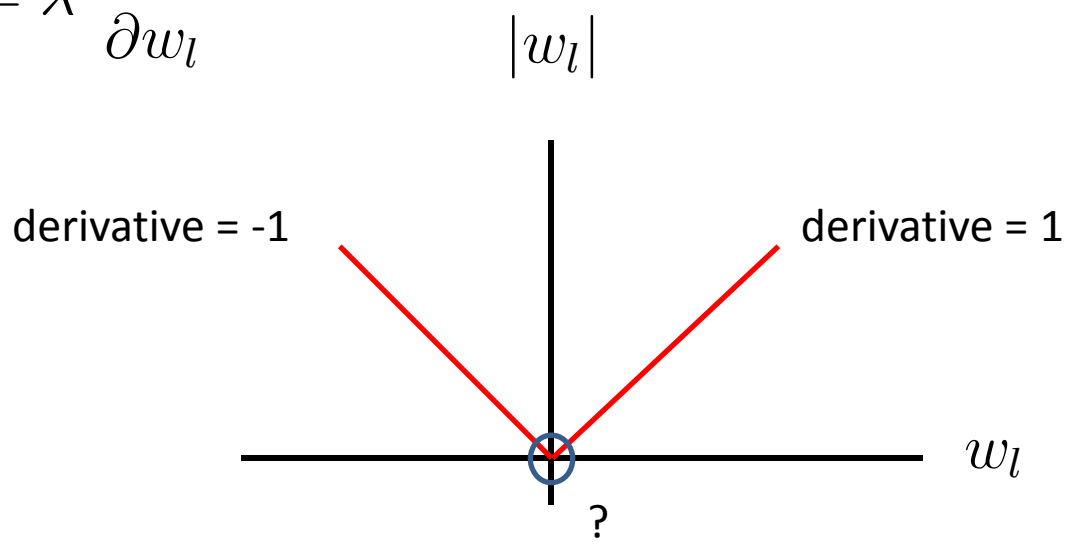
Residual sum of squares (RSS)

$$\frac{\partial}{\partial w_l}RSS(\mathbf{w}) = -\sum_{n=1}^{N}\phi_l\left(\mathbf{x}_n\right)\left[t_n - \mathbf{w}^T\phi(\mathbf{x}_n)\right]$$

Optimizing LASSO objective function

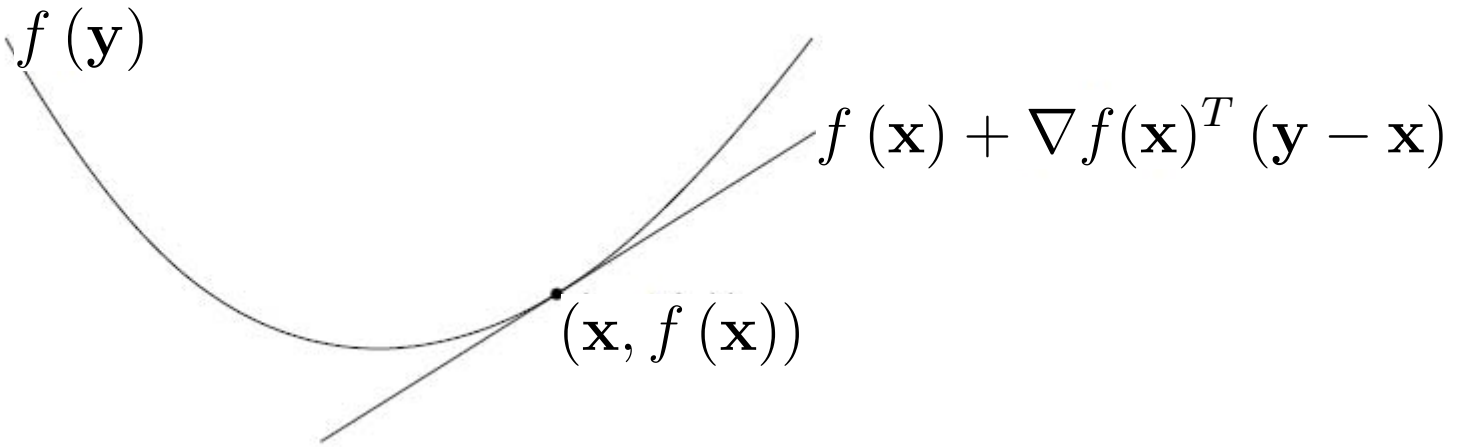
Penalty term:

$$\frac{\partial}{\partial w_l}\lambda\sum_{j=1}^{M}|w_j| = \lambda\frac{\partial\,|w_l|}{\partial w_l}$$



Sub-gradient of convex function

Gradient lower bound of convex function



$$f(\mathbf{x}) + \nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) \leq f(\mathbf{y})$$

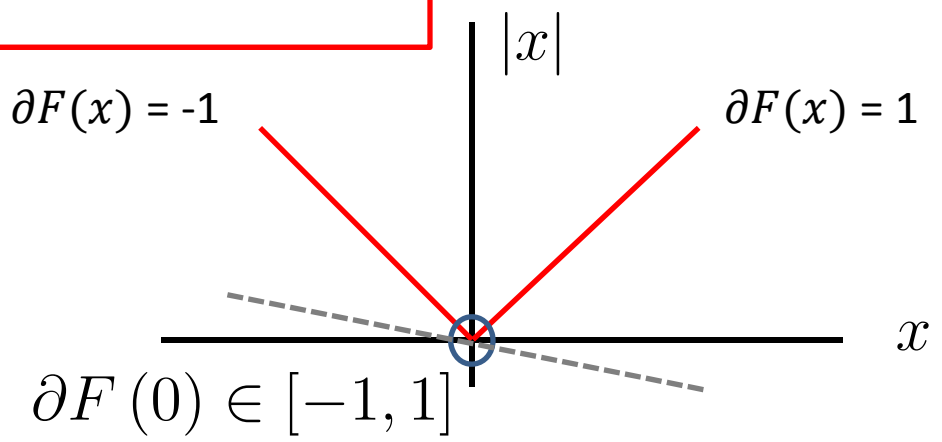
Gradients are unique at **x** if function differentiable at **x**

Sub-gradient of convex function

Sub-gradient: Generalized gradients to non-differentialble points:

Any plane that is lower bound function:

$$F(\mathbf{y}) \geq F(\mathbf{x}) + \partial F(\mathbf{x})^T (\mathbf{y} - \mathbf{x})$$



Taking the sub-gradient

Gradient of RSS term:

$$\frac{\partial}{\partial w_l} RSS(\mathbf{w}) = -\sum_{n=1}^{N} \phi_l(\mathbf{x}_n) \left[t_n - \mathbf{w}^T \phi(\mathbf{x}_n) \right]$$

$$= a_l w_l - c_l$$

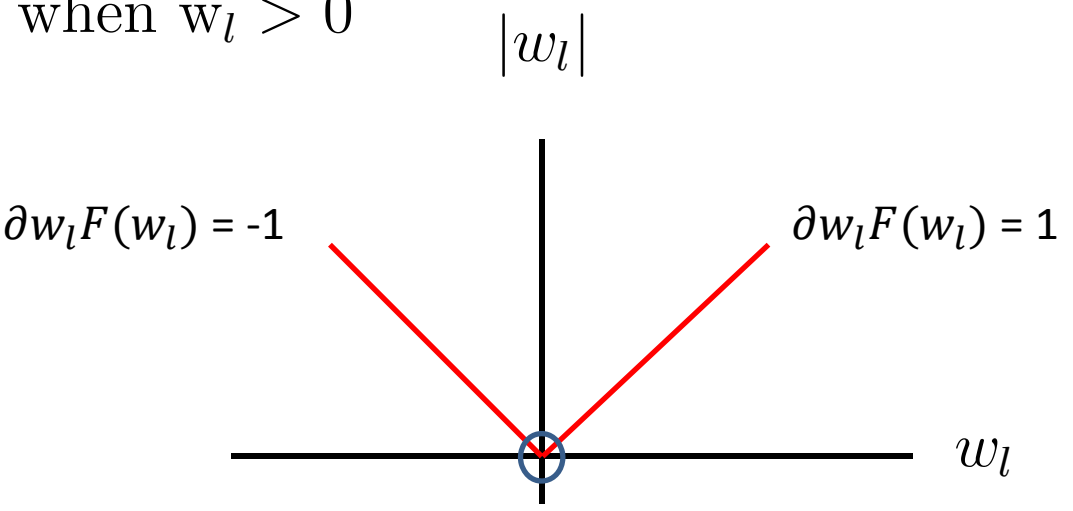
$$a_l = \sum_{n=1}^{N} \phi_l^2(\mathbf{x}_n)$$

$$c_l = \sum_{n=1}^{N} \phi_l(\mathbf{x}_n) \left[t_n - \sum_{l \neq j} w_j \phi_j(\mathbf{x}_n) \right]$$

Taking the sub-gradient

Sub-gradient of penalty

$$\partial_{w_l} |w_l| = \begin{cases} -1 & \text{when } w_l < 0 \\ [-1, 1] & \text{when } w_l = 0 \\ +1 & \text{when } w_l > 0 \end{cases}$$



Taking the sub-gradient

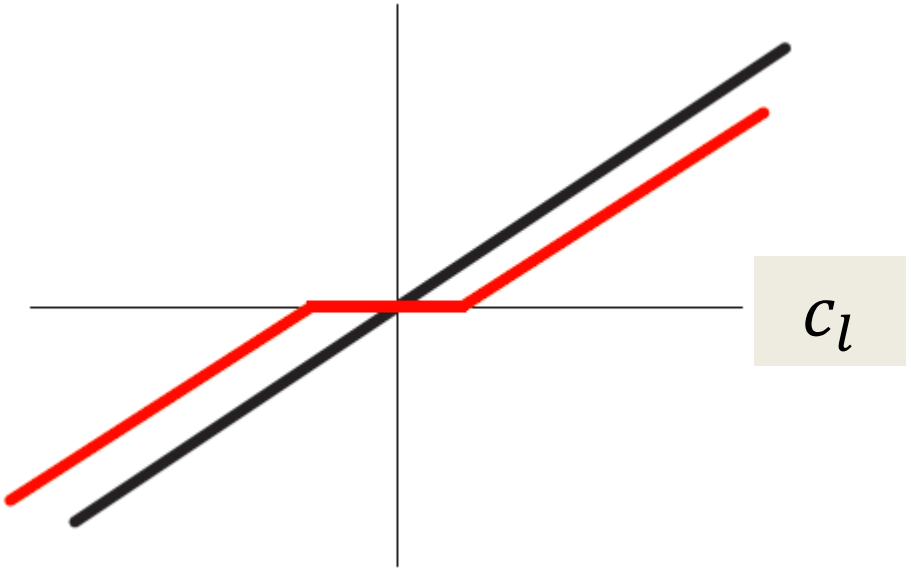
Sub-gradient of full objective function

$$\partial_{w_l} F(\mathbf{w}) = a_l w_l - c_l + \lambda \partial_{w_l} |w_l|$$

$$= \begin{cases} \mathrm{a}_l w_l - c_l - \lambda & \text{when } \mathrm{w}_l < 0 \\ [-c_l - \lambda, -c_l + \lambda] & \text{when } \mathrm{w}_l = 0 \\ \mathrm{a}_l w_l - c_l + \lambda & \text{when } \mathrm{w}_l > 0 \end{cases}$$

Setting subgradient to 0

$$\hat{w}_l = \begin{cases} (\mathrm{c}_l + \lambda)/a_l & \text{when } \mathrm{c}_l < -\lambda \\ 0 & \text{when } \mathrm{c}_l \in [-\lambda, \lambda] \\ (\mathrm{c}_l - \lambda)/a_l & \text{when } \mathrm{c}_l > \lambda \end{cases}$$



Coordinate descent for LASSO

Repeat until convergence

Pick a coordinate l at (random or sequentially)

Set:

$$\hat{w}_l = \begin{cases} (\mathrm{c}_l + \lambda)/a_l & \text{when } \mathrm{c}_l < -\lambda \\ 0 & \text{when } \mathrm{c}_l \in [-\lambda, \lambda] \\ (\mathrm{c}_l - \lambda)/a_l & \text{when } \mathrm{c}_l > \lambda \end{cases}$$

Until convergence

LASSO example

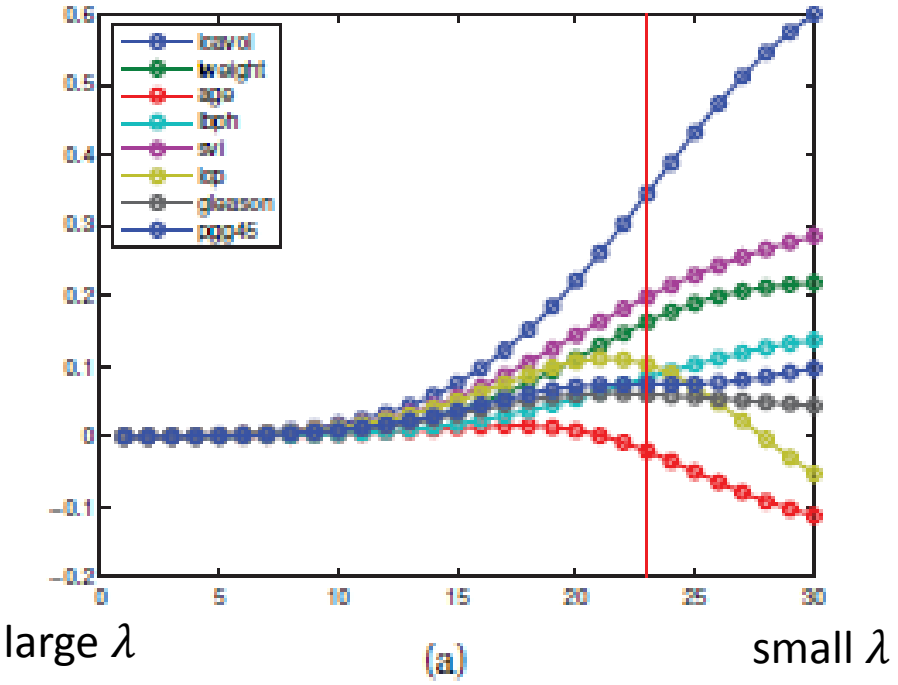
prostate cancer data

8 features and 67 training cases

Term	LS	Best Subset	Ridge	Lasso
Intercept	2.452	2.481	2.479	2.480
lcavol	0.716	0.651	0.656	0.653
lweight	0.293	0.380	0.300	0.297
age	-0.143	-0.000	-0.129	-0.119
lbph	0.212	-0.000	0.208	0.200
svi	0.310	-0.000	0.301	0.289
lcp	-0.289	-0.000	-0.260	-0.236
gleason	-0.021	-0.000	-0.019	0.000
pgg45	0.277	0.178	0.256	0.226
Test Error	0.586	0.572	0.580	0.564

LASSO example

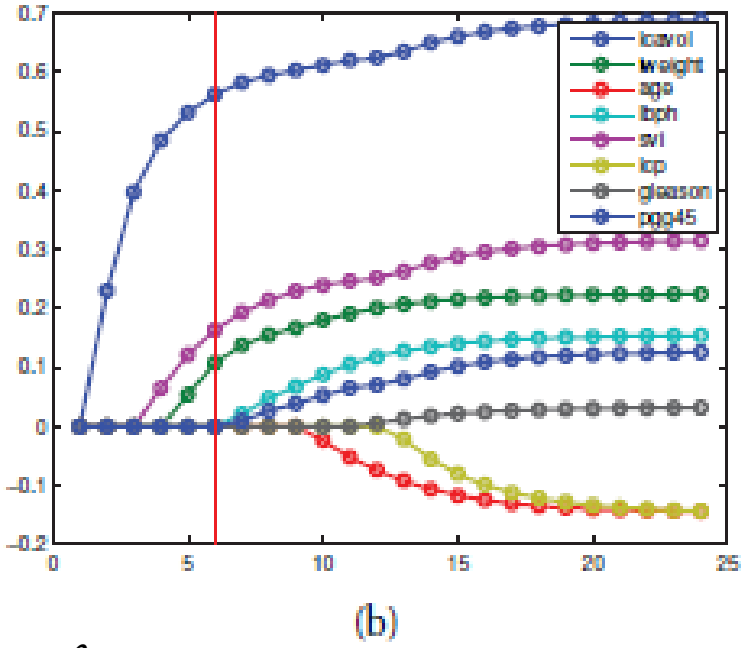
Ridge Regression



large λ small λ

LASSO example

LASSO



(b)

large λ

small λ