Regression

Mark Craven and David Page
Computer Sciences 760
Spring 2018

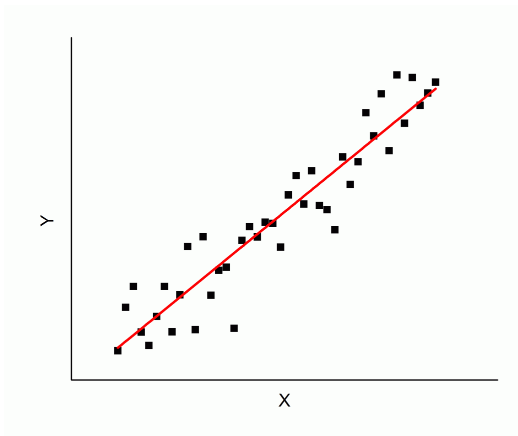
www.biostat.wisc.edu/~craven/cs760

Goals for the lecture

you should understand the following concepts

- linear regression
- RMSE, MAE, and R-square
- ridge regression (L2 penalty)
- Lagrange multipliers
- convex functions and sets
- lasso (L1 penalty): least absolute shrinkage and selection operator
- lasso by proximal method (ISTA)
- lasso by coordinate descent
- logistic regression and penalized logistic regression

Linear Regression



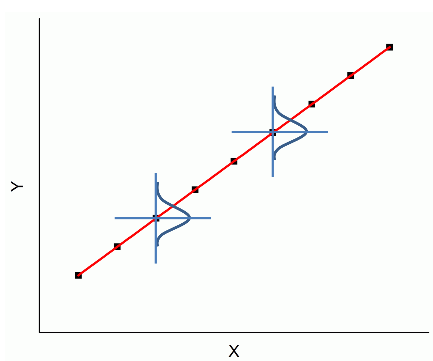
- Linear regression assumes that the relation between the expected value of dependent variable Y and the value of independent variable(s) X, is linear.

Ordinary Least Square (OLS)

- For single variable assume the data is given by

$$y_i = \alpha + \beta x_i + \varepsilon_i$$

where ε_i are Gaussian noises which are independent and have mean 0 and variance σ^2



Many assumptions… Some major ones:

- Linear relationship
  - Can partially address by taking square, cube, exponential, square root, or logarithm of x's or y
  - If modify y, also modifies variance…
- Homoscedasticity (same variance)
- Independence of input features

Other Practicalities

- Might want all features to be distributed as standard normal (Gaussian with mean 0 and standard deviation 1: subtract mean and then divide by standard deviation
  - Simplifies notation, e.g., three slides from now
  - Makes coefficients comparable
- Another option for last sub-bullet: Force values into [0,1] by subtracting Min value and then dividing by Max – Min
- Might pre-compute "interaction terms," e.g., $x_i x_j$: new features to capture non-linearities just like x^2. (So much for third assumption 2 slides ago…)

Ordinary Least Square (OLS)

- Goal: Minimize the objective function:

$$\underset{\text{Objective function}}{error} = \underset{\text{RMSE}}{\sqrt{\sum_i (h(x_i) - y_i)^2}} \text{ or } \underset{\text{MAE}}{\sum_i |h(x_i) - y_i|}$$

- Solution:

$$y = \alpha + \beta x + \varepsilon$$

$$\hat{\beta} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}, \hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$$

Some Terminology You May Hear

- Squared error is one *loss function*
- *Loss function* is a real valued function associating a cost with an outcome (a prediction and actual value pair)
- *Empirical risk* is average loss over training data set
- *Empirical Risk Minimization (ERM)* is a general principle of finding the model in our language with lowest empirical risk

Using Linear Algebra

- As we go to more variables, notation more complex
- Use matrix representation and operations, assume all features standardized (standard normal), and assume an additional constant 1 feature
- Given data matrix **X** with label vector **Y**
- Find vector of coefficients **β** to minimize:
- $||\mathbf{X}\boldsymbol{\beta} - \mathbf{Y}||_2^2$

Multivariate Linear Regression

- Write Matrix X and Y as:

$$X = \begin{pmatrix} \quad & \quad & \quad \end{pmatrix} \quad Y = \begin{bmatrix} \quad \end{bmatrix}$$

- Solution:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

Evaluation Metrics for Numeric Prediction

- Root mean squared error (RMSE)
- Mean absolute error (MAE) – average error
- R-square (R-squared)
- Historically all were computed on training data, and possibly adjusted after, but really should cross-validate

R-square(d)

- Formulation 1:

$$R^2 = 1 - \frac{\sum_i (y_i - h(\vec{x_i}))^2}{\sum_i (y_i - \overline{y})^2}$$

- Formulation 2: square of Pearson correlation coefficient r. Recall for x, y:

$$r = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2}\sqrt{\sum_i (y_i - \overline{y})^2}}$$

Some Observations

- R-square of 0 means you have no model, R-square of 1 implies perfect model (loosely, explains all variation)
- These two formulations agree when performed on the training set
- The do not agree when we do cross-validation, in general, because mean of training set is different from mean of each fold
- Should do CV and use first formulation, but can be negative!

Great things about OLS regression

- Closed-form solution: fast!!
- Works well when given a small number of carefully chosen variables (say < 50)
- Works well even if some assumptions not fully satisfied
- *Models* are understandable
- *Method* is understood by non-stats/ML folks

Estimation of the Warfarin Dose with Clinical and Pharmacogenetic Data

International Warfarin Pharmacogenetics Consortium

(IWPC)

New England Journal of Medicine, February 19, 2009, vol. 360, no. 8

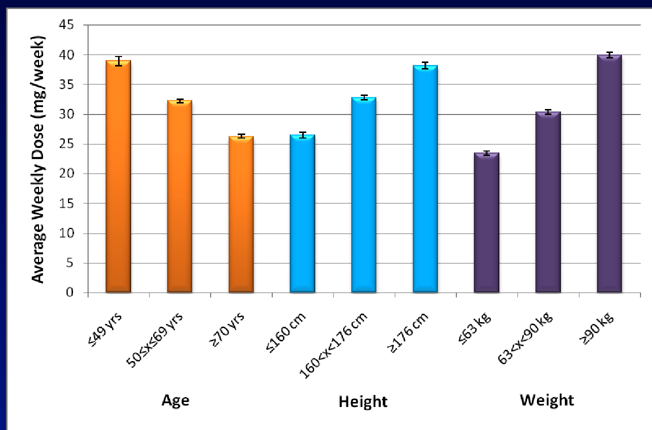
"In Milestone, FDA Pushes Genetic Tests Tied to Drug," *Wall Street Journal, 2007*

Initial dosing (warfarin package insert)

"The dosing of COUMADIN must be individualized according to patient's sensitivity to the drug as indicated by the PT/INR….. It is recommended that COUMADIN therapy be initiated with a dose of 2 to 5 mg per day with dosage adjustments based on the results of PT/INR determinations. The lower initiation doses should be considered for patients with certain genetic variations in CYP2C9 and VKORC1 enzymes as well as for elderly and/or debilitated patients…."

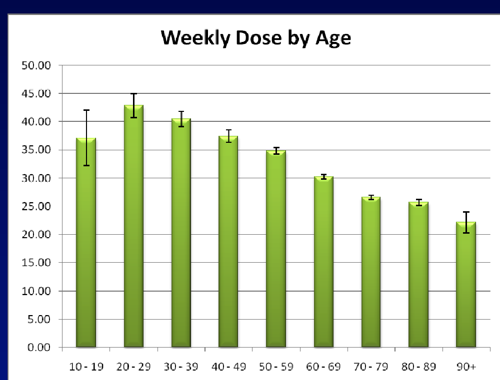
http://www.fda.gov/cder/foi/label/2007/009218s105lblv2.pdf

Age, height and weight



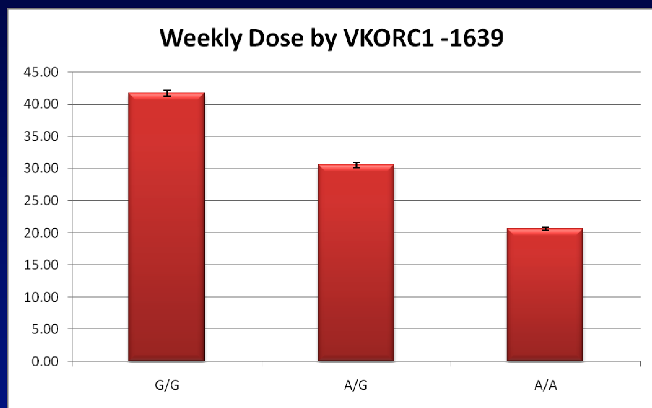
International Warfarin Pharmacogenetics Consortium
iwpc@pharmgkb.org
February 2009

Weekly dose by age



International Warfarin Pharmacogenetics Consortium
iwpc@pharmgkb.org
February 2009

Weekly dose by VKORC1 -1639 genotype



International Warfarin Pharmacogenetics Consortium
iwpc@pharmgkb.org
February 2009

Statistical Analysis

Derivation Cohort
- 4,043 patients with a stable dose of warfarin and target INR of 2-3 mg/week
- Used for developing dose prediction models

Validation Cohort
- 1,009 patients (20% of dataset)
- Used for testing final selected model

Analysis group did not have access to validation set until *after* the final model was selected

International Warfarin Pharmacogenetics Consortium
iwpc@pharmgkb.org
February 2009

Numerical modeling methods used

Included, among others
- Support vector regression
- Regression trees
- Model trees
- Multivariate adaptive regression splines
- Least-angle regression
- Lasso
- Logarithmic and square-root transformations
- Direct prediction of dose

Least-squares linear regression modeling method was best according to criterion yielding the lowest mean absolute error

- Predicted the square root of the dose
- Incorporated both genetic and clinical data

IWPC pharmacogenetic dosing algorithm

**The output of this algorithm must be squared to compute weekly dose in mg

^All references to VKORC1 refer to genotype for rs9923231

		5.6044	
-		0.2614 x	Age in decades
+		0.0087 x	Height in cm
+		0.0128 x	Weight in kg
-		0.8677 x	*VKORC1*^ A/G
-		1.6974 x	*VKORC1* A/A
-		0.4854 x	*VKORC1* genotype unknown
-		0.5211 x	*CYP2C9* *1/*2
-		0.9357 x	*CYP2C9* *1/*3
-		1.0616 x	*CYP2C9* *2/*2
-		1.9206 x	*CYP2C9* *2/*3
-		2.3312 x	*CYP2C9* *3/*3
-		0.2188 x	*CYP2C9* genotype unknown
-		0.1092 x	Asian race
-		0.2760 x	Black or African American
-		0.1032 x	Missing or Mixed race
+		1.1816 x	Enzyme inducer status
-		0.5503 x	Amiodarone status
=	**Square root of weekly warfarin dose****		

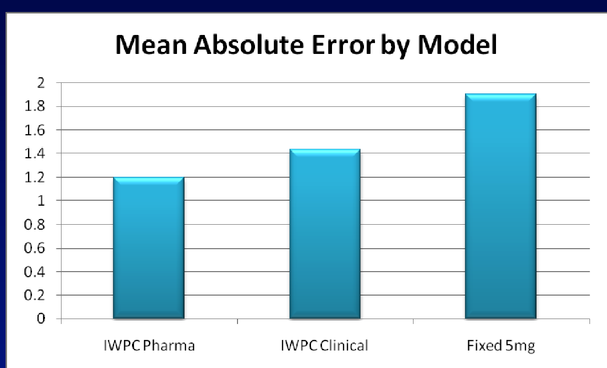
IWPC clinical dosing algorithm

**The output of this algorithm must be squared to compute weekly dose in mg

	4.0376	
-	0.2546 x	Age in decades
+	0.0118 x	Height in cm
+	0.0134 x	Weight in kg
-	0.6752 x	Asian race
+	0.4060 x	Black or African American
+	0.0443 x	Missing or Mixed race
+	1.2799 x	Enzyme inducer status
-	0.5695 x	Amiodarone status
=	**Square root of weekly warfarin dose****	

International Warfarin Pharmacogenetics Consortium
iwpc@pharmgkb.org
February 2009

Model comparisons



International Warfarin Pharmacogenetics Consortium
iwpc@pharmgkb.org
February 2009

Regularized Regression

- Regression is prone to overfitting, especially when:
  - there are a large number of features or
  - It's fit with high order polynomial features
- Regularization helps combat overfitting by having a simpler model. It is used when we want to have:
  - less variation in the different weights or
  - smaller weights overall or
  - only a few non-zero weights (and thus features kept)
- Regularization is accomplished by adding a penalty term to the target function that is being optimized
- Two widely-used types – L_2 and L_1 regularization.

L_2 regularization in linear regression

- What if hundreds or thousands of variables?
- Big risk of overfitting
- Force simpler model, often defined as smaller and "more regular" (less varying) coefficients… small Euclidean norm
- Like limiting maximum decision tree size or depth
- $\underset{\boldsymbol{\beta}}{\text{argmin}} \; ||\mathbf{X}\boldsymbol{\beta} - \mathbf{Y}||_2^2$ such that $||\boldsymbol{\beta}||_2 < s$
- *Constrained* optimization problem… can't just set derivative (gradient) with respect to $\boldsymbol{\beta}$ to 0 and solve as did for OLS

Lagrange Multipliers

To maximize $f(\mathbf{x})$ *such that* $g(\mathbf{x}) < s$

instead maximize: $f(\mathbf{x}) + \lambda(g(\mathbf{x}) - s)$

- λ is *Lagrange multiplier*
- Resulting optimization task is *unconstrained*

- To find $\boldsymbol{\beta}$ to minimize $||\mathbf{X}\boldsymbol{\beta} - \mathbf{Y}||_2^2$ s.t. $||\boldsymbol{\beta}||_2 < s$:
- find $\boldsymbol{\beta}$ to minimize $||\mathbf{X}\boldsymbol{\beta} - \mathbf{Y}||_2^2 + \lambda\,(||\boldsymbol{\beta}||_2 - s)$
- In practice since we tune hyperparameter λ, s doesn't matter, so problem becomes:
  find $\boldsymbol{\beta}$ to minimize $||\mathbf{X}\boldsymbol{\beta} - \mathbf{Y}||_2^2 + \lambda||\boldsymbol{\beta}||_2$

L_2 regularization in linear regression

- Called "ridge regression"
- Still has a closed-form solution, so even though continuous differentiable and convex, don't need gradient descent
- Setting gradient with respect to β, from previous slide, to 0 and solving we get:
- $\beta = (X^TX - \lambda I)^{-1}X^TY$

Simple Lagrange Multipliers Example (Thanks Wikipedia!)

Minimize $f(x,y) = x + y$ such that $x^2 + y^2 = 1$

Note that constraint is: $g(x,y) = x^2 + y^2 - 1$

$$\begin{aligned}\mathcal{L}(x,y,\lambda) &= f(x,y) + \lambda \cdot g(x,y) \\ &= x + y + \lambda(x^2 + y^2 - 1).\end{aligned}$$

Now we can calculate the gradient:

$$\begin{aligned}\nabla_{x,y,\lambda}\mathcal{L}(x,y,\lambda) &= \left(\frac{\partial \mathcal{L}}{\partial x}, \frac{\partial \mathcal{L}}{\partial y}, \frac{\partial \mathcal{L}}{\partial \lambda}\right) \\ &= \left(1 + 2\lambda x, 1 + 2\lambda y, x^2 + y^2 - 1\right)\end{aligned}$$

$$\nabla_{x,y,\lambda}\mathcal{L}(x,y,\lambda) = 0 \quad \Leftrightarrow \quad \left.\begin{cases} 1 + 2\lambda x = 0 \\ 1 + 2\lambda y = 0 \\ x^2 + y^2 - 1 = 0 \end{cases}\right\} \text{Read as AND}$$

The first two equations yield

$$x = y = -\frac{1}{2\lambda}, \qquad \lambda \neq 0.$$

By substituting into the last equation we have:

$$\frac{1}{4\lambda^2} + \frac{1}{4\lambda^2} - 1 = 0,$$

so

$$\lambda = \pm\frac{1}{\sqrt{2}},$$

which implies that the stationary points of $\mathcal{L}$ are

$$\left(\tfrac{\sqrt{2}}{2}, \tfrac{\sqrt{2}}{2}, -\tfrac{1}{\sqrt{2}}\right), \qquad \left(-\tfrac{\sqrt{2}}{2}, -\tfrac{\sqrt{2}}{2}, \tfrac{1}{\sqrt{2}}\right).$$

Can work out that the constrained maximum is $\sqrt{2}$

Logistic Regression: Motivation

- Linear regression was used to fit a linear model to the feature space in order to predict continuous response
- Suppose response is *binary;* predict positive if linear function exceeds some value: step function
- But also want to produce a probability that a feature will take a particular value given other features

$$P(Y = 1 \mid X)$$

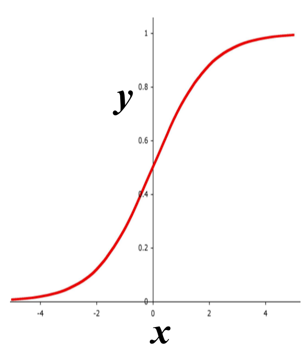
- So, extend linear regression for classification; no closed-form solution anymore, so need to do *gradient descent*

Logistic (Sigmoid) Function

- To exhibit the relation between a dependent and an independent variable, we could use a step function. But it is not differentiable.
- We need a continuous and differentiable function: ***Logistic function***

$$y = \frac{1}{1+e^{-cx}}$$
$$= \frac{1}{1+e^{-wx}}$$

wx →a linear function of the feature vector x



The Algorithmic Approach

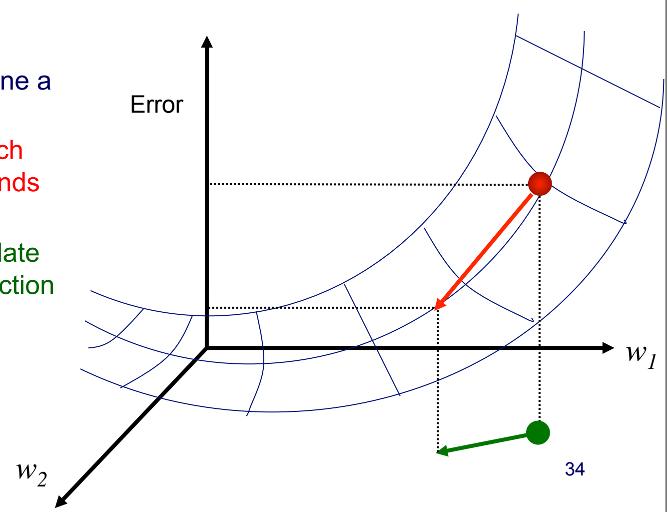
- Instead of squared error, want to minimize probability (according to model) of incorrect class
- So error E is 1 – probability of correct class
- Probability of data according to model is likelihood of model; probability of correct class is *conditional likelihood* (more on likelihood in Bayes nets)
- No closed form solution for **w**; we will have to rely on *gradient descent* to minimize E (more in neural nets)

Gradient descent in weight space

gradient descent is an iterative process aimed at finding a minimum in the error surface

on each iteration

- current weights define a point in this space
- find direction in which error surface descends most steeply
- take a step (i.e. update weights) in that direction



34

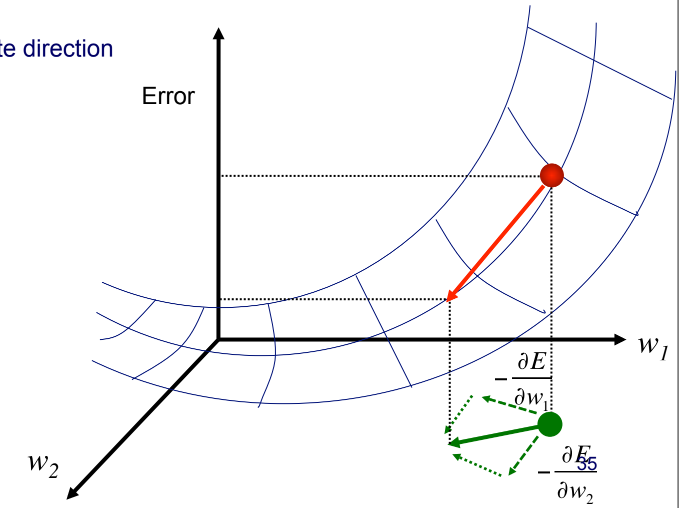
Gradient descent in weight space

calculate the gradient of E: $\nabla E(\boldsymbol{w}) = \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \cdots, \frac{\partial E}{\partial w_n}\right]$

take a step in the opposite direction

$$\Delta \boldsymbol{w} = -\eta \, \nabla E(\boldsymbol{w})$$

$$\Delta w_i = -\eta \, \frac{\partial E}{\partial w_i}$$



Logistic Regression Algorithm

- The conditional log likelihood is given by
  $l(w) = \ln(\prod_j P(y_{(j)} \mid \boldsymbol{x}_{(j)}, \boldsymbol{w}))$, *where j → jth sample*
- Need to find 'w' that maximizes the conditional log likelihood
  $\underset{w}{\arg\max} \ \ln(\prod_j P(y_{(j)} \mid \boldsymbol{x}_{(j)}, \boldsymbol{w}))$
- Can use gradient ascent
  $w_i^{new} := w_i + \eta \frac{\partial l(\boldsymbol{w})}{\partial w_i}$, *where η → learning rate parameter, i → i^{th} feature*
- The derivative comes out to:
  $\frac{\partial l(\boldsymbol{w})}{\partial w_i} = \sum_j x_{i_{(j)}} \left(y_{(j)} - \mathrm{P}\left(y_{(j)} = 1 \,\middle|\, \boldsymbol{x}_{(j)}, \boldsymbol{w}\right)\right)$
- This gives us the gradient ascent rule:

$$w_i^{new} := w_i + \eta \sum_j x_{i_{(j)}} \underbrace{\left(y_{(j)} - \mathrm{P}(y_{(j)} = 1 \mid \boldsymbol{x}_{(j)}, \boldsymbol{w})\right)}_{\text{Error in estimate}}$$

More on Gradient Descent

- Gradient descent yields an optimal solution if the minimization problem is *convex*
- Can compute gradient at once over all examples (batch) or compute from one example at a time (*stochastic* gradient descent, where stochastic part is next example randomly chosen)

Convexity (from Bubeck, 2015)

Definition 1.1 (Convex sets and convex functions)**.** A set $\mathcal{X} \subset \mathbb{R}^n$ is said to be convex if it contains all of its segments, that is

$$\forall (x, y, \gamma) \in \mathcal{X} \times \mathcal{X} \times [0,1], \ (1-\gamma)x + \gamma y \in \mathcal{X}.$$

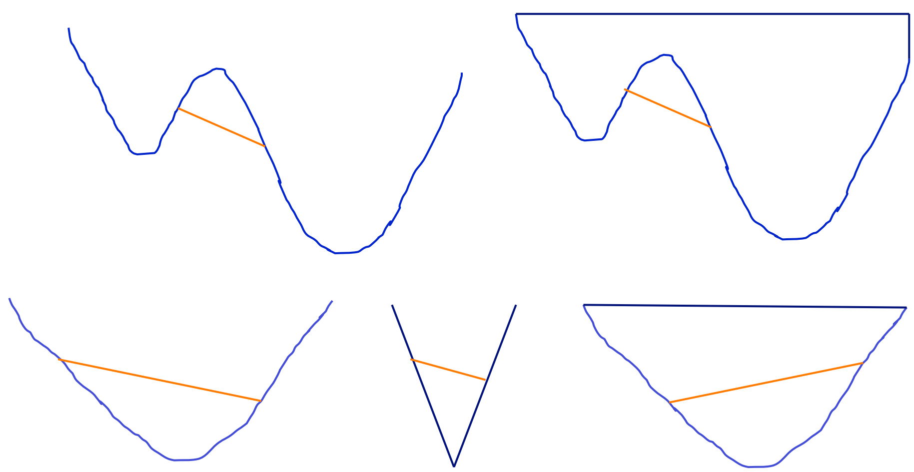
A function $f : \mathcal{X} \to \mathbb{R}$ is said to be convex if it always lies below its chords, that is

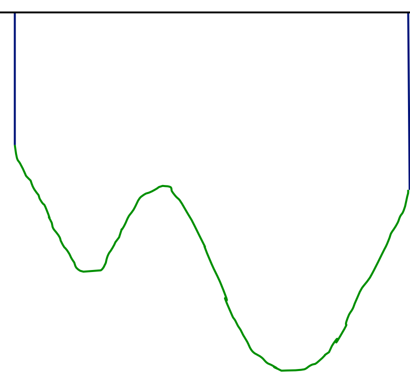
$$\forall (x, y, \gamma) \in \mathcal{X} \times \mathcal{X} \times [0,1], \ f((1-\gamma)x + \gamma y) \leq (1-\gamma)f(x) + \gamma f(y).$$

We are interested in algorithms that take as input a convex set $\mathcal{X}$ and a convex function f and output an approximate minimum of f over $\mathcal{X}$. We write compactly the problem of finding the minimum of f over $\mathcal{X}$ as

$$\begin{aligned} &\text{min. } f(x) \\ &\text{s.t. } x \in \mathcal{X}. \end{aligned}$$

(Non-)Convex Function or Set





Epigraph (Bubeck, 2015)

$$\text{epi}(f) = \{(x, t) \in \mathcal{X} \times \mathbb{R} : t \geq f(x)\}.$$

It is obvious that a function is convex if and only if its epigraph is a convex set.

- Show for all real $a < b$ and $0 \le c \le 1$,
  $f(ca + (1-c)b) \le c\, f(a) + (1-c)\, f(b)$ for following:
  - $f(x)=|x|$
  - $f(x)=x^2$
  - Not so for $f(x)=x^3$
- In general x could be a vector **x**
- For gradient descent, also want f(x) to be continuous differentiable
- For |x| we need proximal methods, subgradient methods, or coordinate descent

Comments on basic logistic regression

- Logistic Regression is a linear classifier
- Logistic Regression optimized by using conditional likelihood
  - no closed-form solution
  - Error function is continuous differentiable – can always compute gradient
  - *convex* -> find global optimum with gradient ascent

L_1 regularization

- L1 regularization uses 1-norm of the weight vector in the penalty term as shown:

  $\lambda||w||_1$

  i.e., $\lambda \sum_i |w|_i$ *where 'i' represents the i^{th} feature*
- Also called 'Lasso' penalty.
- Gradient ascent is no longer feasible since L1 norm is not differentiable.

LASSO: Penalty as a Constraint

Add penalty as a constraint to OBJ function:

Find $\hat{\alpha}$ *and* $\hat{\beta}$

To minimize $\underbrace{\sum_i (h(\vec{x_i}) - y_i)^2}_{error}$

Such that $\underbrace{\sum_j |\hat{\beta}_j| \leq s}_{constraint}$ $\quad$ *(s is a constant)*

LASSO: Penalty as a Term in OBJ

Add penalty as a term to OBJ function to be minimized:

(λ is penalty parameter)

$$\underbrace{\sum_i (h(\vec{x_i}) - y_i)^2}_{error} + \lambda \underbrace{\sum_j |\hat{\beta}_j|}_{penalty}$$

Obtained by taking Langrangian. Even for linear regression, no closed-form solution. Ordinary gradient ascent also does not work because no derivative. Fastest methods now FISTA and (faster) coordinate descent.

Proximal Methods

- *f(x) = g(x) + h(x)*
  - *g* is convex, differentiable
  - *h* is convex and decomposable, but not differentiable
  - Example: g is squared error, h is lasso penalty – sum of absolute value terms, one per coefficient (so one per feature)
  - Find $\boldsymbol{\beta}$ to minimize $\underbrace{||\mathbf{X}\boldsymbol{\beta} - \mathbf{Y}||_2^2}_{g} + \underbrace{\lambda||\boldsymbol{\beta}||_1}_{h}$

Proximal Operator: Soft-Thresholding

$S_\lambda(\boldsymbol{x})$ =
for all i

$$\begin{cases} x_i - \lambda & \text{if } x_i > \lambda \\ 0 & \text{if } -\lambda \le x_i \le \lambda \\ x_i + \lambda & \text{if } x_i < -\lambda \end{cases}$$

We typically apply this to coefficient vector β.

Iterative Shrinkage-Thresholding Algorithm (ISTA)

- Initialize $\hat{\beta}$; let η be learning rate
- Repeat until convergence
  - Make a gradient step of:

    $\boldsymbol{\beta} \leftarrow \quad S_\lambda(\boldsymbol{\beta} - \eta \mathbf{X}^T(\mathbf{X}\boldsymbol{\beta} - \mathbf{y}))$

Coordinate Descent

- *Fastest* current method for lasso-penalized linear or logistic regression
- *Simple* idea: adjust one feature at a time, and special-case it near 0 where gradient not defined (where absolute value's effect changes)
- Can take features in a cycle in any order, or randomly pick next feature (analogous to Gibbs Sampling)
- To "special-case it near 0" just apply soft-thresholding everywhere

Coordinate Descent Algorithm

- Initialize coefficients
- Cycle over features until convergence:
  - For each example *i* and feature *j*, compute "partial residual":

$$r_{ij} = y_i - \sum_{k \neq j} x_{ik} \beta_k$$

  - Compute least-squares coefficients of these residuals (as we did in OLS regression):

$$\beta_j^* = \frac{1}{N} \sum_{i=1}^{N} x_{ij} r_{ij}$$

  - Update β_j by soft-thresholding, where for any term T, "T_+" denotes min(0,A):

$$\beta_j \leftarrow S_\lambda(\beta_j^*)$$

Comments on penalized regression

- L2-penalized regression also called "ridge regression"
- Can combine L1 and L2 penalties: "elastic net"
- L1-penalized regression is especially active area of research
  - group lasso
  - fused lasso
  - others

L_2 regularization in logistic regression

- L_2 regularization uses 2-norm of the weight vector in the penalty term as shown:

  $\lambda||w||_2^2$

  i.e., $\lambda \sum_i w_i^2$ *where 'i' represents the ith feature*
- 'λ' is the regularization parameter used to control the weights. It governs how big the penalty is relative to fitting the data well.
- When the penalty term is added to the objective function, the gradient ascent algorithm's update changes to:

$$w_i^{new} := w_i + \eta \sum_j x_{i_{(j)}} \left(y_{(j)} - \mathrm{P}\left(y_{(j)} = 1 \,\middle|\, x_{(j)}\, w \right) \right) - \eta \lambda w_i$$

where, $'\eta'$ is the learning rate parameter

More comments on regularization

- Linear and logistic regression prone to overfitting
- Regularization helps combat overfitting by adding a penalty term to the target function being optimized
- L1 regularization often preferred since it produces sparse models. It can drive certain co-efficients(weights) to zero, performing feature selection in effect
- L2 regularization drives towards smaller and simpler weight vectors but cannot perform feature selection like L1 regularization
- Few uses of OLS these days… e.g., Warfarin Dosing (NEJM 2009)… just 30 carefully hand-selected features