Feature Selection in Linear Models

Abhishek Jaiantilal

April 2, 2008

Feature Selection

- In ML, we get data from different sources, many of them might be noisy.
- Popular methods are based on building a regression or a classification model out of these source data.
- Noisy data causes lower accuracy.

Feature Selection (crude definition)

Finding out what sources are more relevant for predicting. Thus also implying we find sources that are noisy (and contribute less & and not use them in model).

Another motivation: Learning from few examples

Say you have very less examples and each example has many features(in thousand). (e.g. vision data where you can have many different features for each pixel like color, texture, filters etc. Other examples include text data). *How cool it would be if you can make a model that has good prediction accuracy based on few examples?*

Regression Notation

Let's Start with the notation

$$y = X\beta \tag{1}$$

Where y = nx1 (known)
X = nxd (known)
β = dx1 (to estimate)

'n' examples, **'d'** dimensions.

Equates

$$Y_1 = X_{11}\beta_1 + ..X_{1d}\beta_d$$

$$...$$

$$Y_n = X_{n1}\beta_1 + ..X_{nd}\beta_d$$

Some more Notation

Note

In this Lecture: Betas, Features, Variables, Predictors all refer to β's that is weighing of the dimension(d).

$$Y_1 = X_{11}\beta_1 + ..X_{1d}\beta_d$$

$$...$$

$$Y_n = X_{n1}\beta_1 + ..X_{nd}\beta_d$$

if $n = d$ determined.
if $n > d$ overdetermined.
if $n < d$ underdetermined. (I will be talking about this today)

Non-perfect ML world

- Due to errors we don't get an exact fit for y=Xβ
- In ML we assume that the error induced is Zero mean & finite variance σ.
- So instead of response Y, we get an estimate/predicted response $\hat{Y}$

Equates

Residual,

$$\begin{aligned} r &= Y - \hat{Y} \\ &= Y - X\beta \end{aligned}$$

Metric for choosing β Estimates

- How do you know that your β estimates are good?
- Use a metric on your residual $f(Y - \hat{Y}) = f(Y - X\beta)$
- Popular metric M.S.E $= ||Y - X\beta||^2$

Why?

Why use least squares for the error, and not some other norm?

Metric for choosing β Estimates

- How do you know that your β estimates are good?
- Use a metric on your residual $f(Y - \hat{Y}) = f(Y - X\beta)$
- Popular metric M.S.E $= ||Y - X\beta||^2$

Why?

Why use least squares for the error, and not some other norm?

- Least Squares is BLUE (**B**est **L**inear **U**nbiased **E**stimator).

Fact

Gauss markov Theorem: if the error is Zero mean and constant variance among all observation then Least Squares is BLUE.

- Note: Unbiased means that E(β) $= \beta$. (Expected value of β found is equal to the real β)

Stop and Ask

Any Questions Yet?

We have seen why MSE is a good choice for a loss function.

Let's now see how to find the β estimates for M.S.E.

Finding β

- Differentiate $||Y - X\beta||^2$ w.r.t to β & equate to 0.
- $$\beta = (X^T X)^{-1} X^T Y \tag{2}$$
- Any Problems with this formulation?

Finding β

- Differentiate $||Y - X\beta||^2$ w.r.t to β & equate to 0.
- $$\beta = (X^T X)^{-1} X^T Y \tag{2}$$
- Any Problems with this formulation?
- If columns in X are collinear then $X^T X$ is close to singular. (e.g. get measurement of weights in Kgs & Lbs.)
- Thus is $X^T X$ is non-invertible then unique estimator β is non-existent.
- Least Squares Solutions for such Ill-posed problems might have large values of β's and small pertubations will have larger effects.

Solutions?

- What can one do to reduce this (other than Ridge Regression)?

Solutions?

- What can one do to reduce this (other than Ridge Regression)?
- Something (pre-processing) that you did with your homework data.

Solutions?

- What can one do to reduce this (other than Ridge Regression)?
- Something (pre-processing) that you did with your homework data.
- **Standardizing**: Zero mean and Std. Deviation of 1 for each column in X.
- It reduces the effects but not entirely.

Ridge regression

$$minimize||Y - X\beta||^2 + \lambda||\beta||^2 \quad (3)$$

Solve for various values of λ

$$||Y - X\beta||^2 \quad (Loss)$$
$$\lambda||\beta||^2 \quad (Penalty)$$

Solution : $X = (X^T X + \lambda I)^{-1} X^T Y$
Generalized Inverse, Tikhonov Regularization, Pseudoinverse ....
λI helps overcome ill-posed problems.

Derivation of Pseudo Inverse for Ridge Regression

$$||Y - X\beta||^2 + \lambda||\beta||^2$$

$$\begin{aligned} &= Y^T Y - 2Y^T X\beta + \beta^T X^T X\beta + \lambda\beta^T\beta \\ &= \beta^T(X^T X + \lambda I)\beta - 2Y^T X\beta + Y^T Y \end{aligned}$$

let, $X^T X + \lambda$ I = P

$$\begin{aligned} &= \beta^T P\beta - 2Y^T X\beta + Y^T Y \\ &= (\beta^T\beta - \frac{2Y^T X\beta}{P} + \frac{Y^T Y}{P})P \\ &= (\underbrace{\beta^T\beta - \frac{2Y^T X\beta}{P} + \frac{Y^T XX^T Y}{P^2}} - \frac{Y^T XX^T Y}{P^2} + \frac{Y^T Y}{P})P \\ &= (\underbrace{(\beta - \frac{X^T Y}{P})^2} - \frac{Y^T XX^T Y}{P^2} + \frac{Y^T Y}{P})P \end{aligned}$$

above is *minimum at* $\beta = \frac{X^T Y}{X^T X + \lambda I}$

Unbiasness & Ill-posed problems

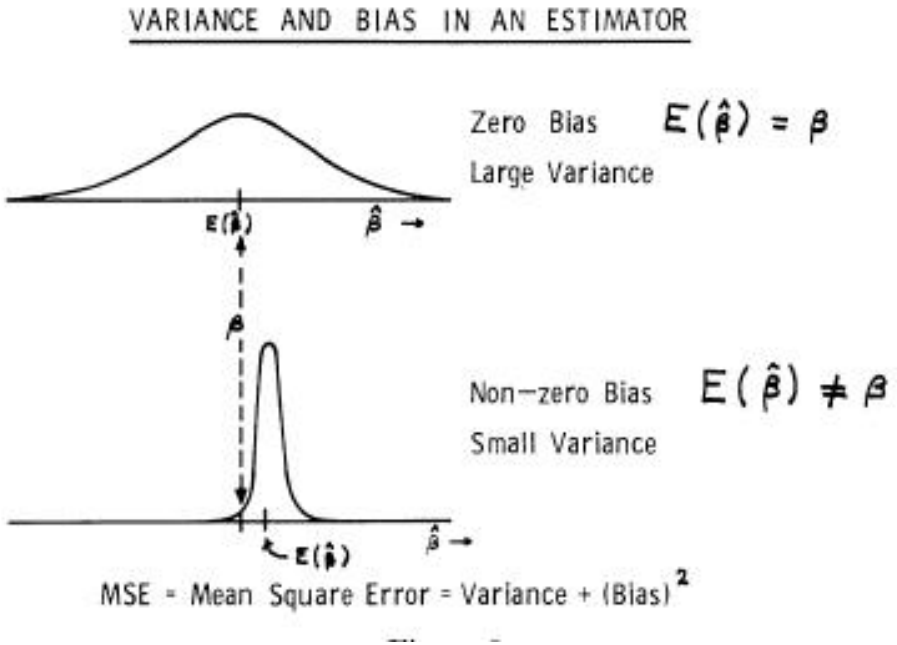


Figure: Least squares Regression vs. Ridge Regression. (Marquart & Snee. Ridge Regression in Practice '75 [3])

- (Earlier)Zero bias but large variance.
- Bind the magnitudes of β's. Thereby variance decreases but at the expense of biasness.

Ridge parameter effects on Betas

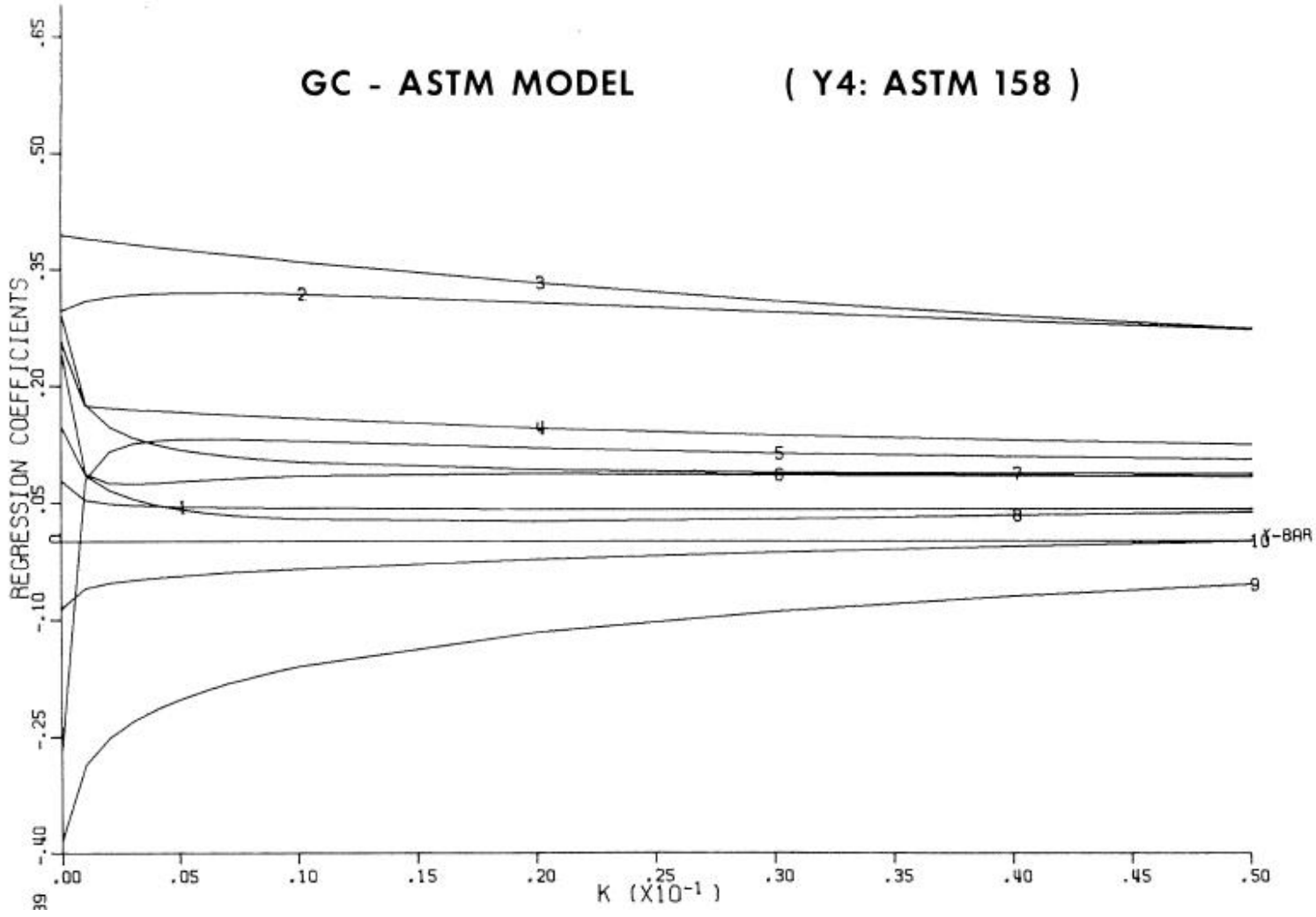


Figure: Betas(Y axis) Vs. Ridge parameter(X axis) (Marquart & Snee. Ridge Regression in Practice '75 [3])

Stop and Ask

Any Questions Yet?

- We have seen Ridge Regression sacrifices bias to get low variance and in a way get around ill-posed questions.
- Introduced the Loss+Penalty form for optimization in ML.
- Let's now see whether we can replace $\lambda||\beta||^2$ with some other penalty function.

What are some penalty functions?

- L_0 = count of all non-zero values (N.P. Complete)
- L_1 = Euclidean norm ($||x||_1 = |x_1| + |x_2| + ... + |x_n|$)
- L_2 = Manhattan norm ($||x||_2 = \sqrt{|x_1|^2 + |x_2|^2 + ... + |x_n|^2}$)
- L_∞ = max of all

We will cover L_1 & L_2

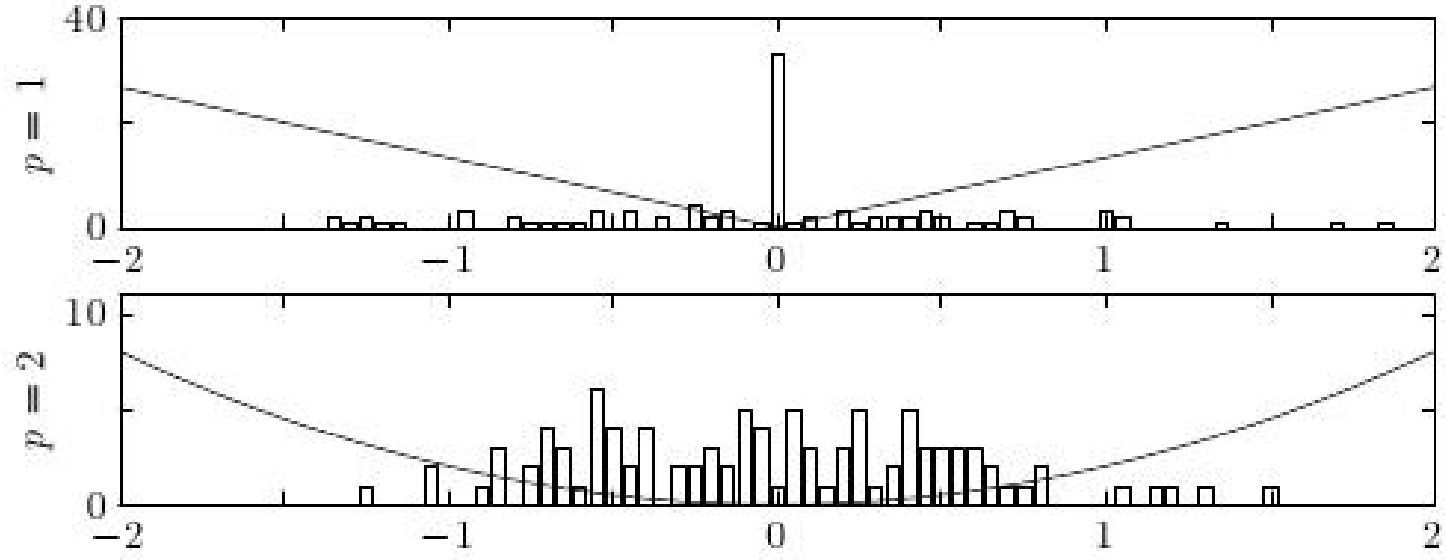


Figure: Histogram of Penalty functions. (Boyd & Vanderberghe. Convex Optimization '04 [1]). p=norm

- L_1 penalty encourages large magnitudes but sparser solution
- L_2 penalty encourages smaller magnitudes but a non-sparse solution(compared to L_1)

Till Now...

- Why we use Mean Square Error Loss (for regression)
- What property one should look in the Penalty function

Feature Selection

What?

Feature Selection is the selection of relevant features (β) when the data has a mix of relevant and irrelevant features.

Assuming that the loss function doesn't introduce selection, what penalty function will introduce sparseness?

Ans: L_1 (Ideally L_0, but that is N.P. Complete)

Ways to solve L_1 penalty (IMHO)?

- How to solve L_1 penalty based problem?
- 1. Pose as an Convex Optimization problem.
- 2. Identify certain properties and optimize on the basis of that.

Lasso. (Tibishirani '96 [7])

$$minimize||Y - X\beta||^2 + \lambda||\beta|| \tag{4}$$

- Doesn't have a nice pseudo-inverse like solution.
- Usually solved by framing as an Convex Optimization Problem.
- Equations 4 & 5 are Equivalent:

$$minimize||Y - X\beta||^2 \; subject \; to \; ||\beta|| \leq t \tag{5}$$

- Slow due to posing as an Convex Optimization problem.

Stop and Ask

Any Questions Yet?

Introduction for the next section

- The next some ideas are based on regression analysis going way back to the 70's.
- 'Subset selection Methods'.

Stepwise Methods

- Idea of Feature Selection is to choose a subset of features to explain the whole model.
- Another way to look at Feature Selection is that we should try to include only relevant Beta's in the model!
- So why don't we add Betas one by one to the model and make it more complex?
- This is the exact idea behind many Subset selection methods like Stepwise methods.

Stepwise Methods

- Forward selection(FS)
  - Begin with single feature that has biggest correlation with Y.
  - Add to the model that has highest criterias like correlation, R^2, or F-statistics, etc.
- Backward elimination(BE)
  - Begin with Full model.
  - Delete from the model, feature that has smallest among the criterias.
- Stepwise selection(SW)
  - Allows Adding/ Deleting/ Exchaning of vairables. (basically a mix of FS and BE)/
- Refer to Weisberg (Applied Linear regression) for details.
- Why correlation? Because it's the Gradient of MSE.

Problems with Stepwise methods

Problems:

- Might not be the best subset. Just having another feature in the model might produce totally different models.
- Too much variability in the model, as it tend to include the whole feature and move the (the already included)predictors according to it.
- Best when many of the variables are uncorrelated. But then finding the subset model is not relevant(as we would then require all of the variables in the final model!)

Stop and Ask

Any Questions Yet?

Forward Stagewise [5]

- Instead of adding the feature totally. Add the feature and move in that direction slowly.
- Algorithm:
  - Begin with the feature 'j' most correlated with residual r. ($r = y - \hat{y}$, initially r=y)
  - Update $\beta_j = \beta_j + \delta_j$ where $\delta_j = \epsilon * sign(correlation(r, x_j))$
  - Update $r = r - \delta_j x_j$ and repeat the previous and this step till no predictor has any correlation with r.

Comparison to Stepwise Methods

- Better models as we move slowly in the direction of the predictors.
- Comparatively slower.

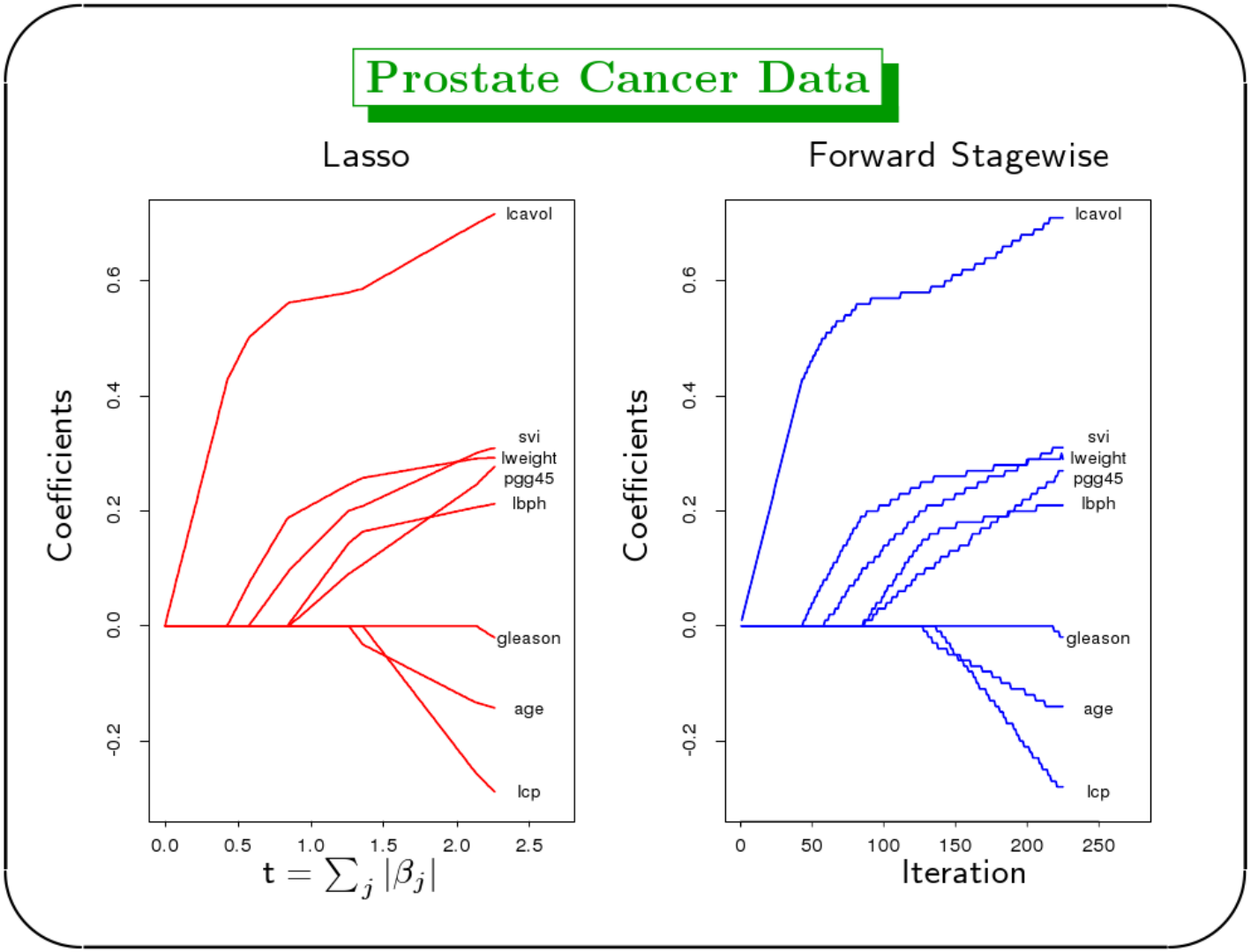


Figure: Forward Stagewise Vs. Lasso (Trevor Hastie. Lars Talk)

Why are Forward Stagewise and Lasso so similar?

- Are they identical?
- In orthogonal predictor case: *yes*
- In hard to verify case of *monotone* coefficient paths: *yes*
- In general, almost!
- Least angle regression (LAR) provides answers to these questions, and an efficient way to compute the complete Lasso sequence of solutions.

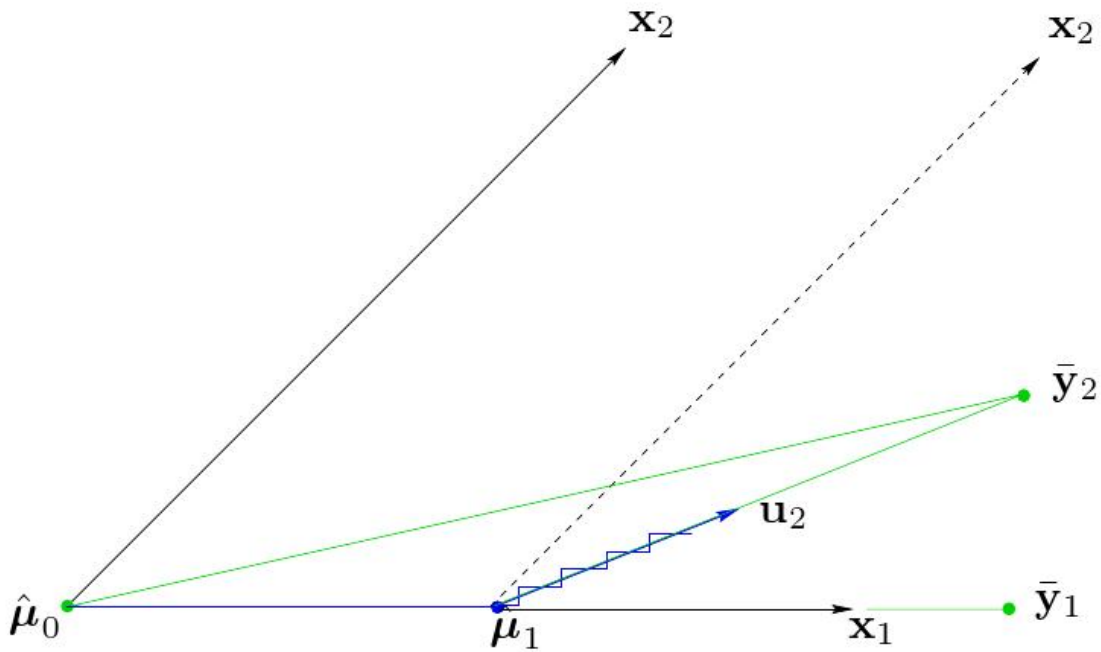
From Trevor Hastie's Lar's Talk

Least Angle Regression — LAR

Like a "more democratic" version of forward stepwise regression.

1. Start with $r = y$, $\hat{\beta}_1, \hat{\beta}_2, \ldots \hat{\beta}_p = 0$. Assume x_j standardized.
2. Find predictor x_j most correlated with r.
3. Increase β_j in the direction of $\text{sign}(\text{corr}(r, x_j))$ until some other competitor x_k has as much correlation with current residual as does x_j.
4. Move $(\hat{\beta}_j, \hat{\beta}_k)$ in the joint least squares direction for (x_j, x_k) until some other competitor x_ℓ has as much correlation with the current residual
5. Continue in this way until all predictors have been entered. Stop when $\text{corr}(r, x_j) = 0 \; \forall \; j$, i.e. OLS solution.

From Trevor Hastie's Lar's Talk



The LAR direction $\mathbf{u}_2$ at step 2 makes an equal angle with $\mathbf{x}_1$ and $\mathbf{x}_2$.

From Trevor Hastie's Lar's Talk

Relationship between the 3 algorithms

- Lasso and forward stagewise can be thought of as restricted versions of LAR
- *For Lasso*: Start with LAR. If a coefficient crosses zero, stop. Drop that predictor, recompute the best direction and continue. This gives the Lasso path

Proof (lengthy): use Karush-Kuhn-Tucker theory of convex optimization. Informally:

$$\frac{\partial}{\partial \beta_j}\{||\mathbf{y}-\mathbf{X}\beta||^2 + \lambda\sum_j |\beta_j|\} = 0$$

$$\Leftrightarrow$$

$$\langle \mathbf{x}_j, \mathbf{r}\rangle = \frac{\lambda}{2}\text{sign}(\hat{\beta}_j) \quad \text{if } \hat{\beta}_j \neq 0 \text{ (active)}$$

From Trevor Hastie's Lar's Talk

Stop and Ask

Any Questions Yet?

- We have seen that there are Efficient ways to solve for L_1 penalty. Property being that the paths are piecewise linear.
- LARS [4] can compute for both Lasso and Stagewise Regression with slight modifications. (Regression)
- Next: Lets see if we can use some other Loss function including for SVM's and Robust regression.
- But before that let's see what are the conditions that make this a possibility?

Piecewise Linear Regularized Paths Rosset & Zhu [6]

$$\hat{\beta}(\lambda) = min\ L(Y, X\beta) + \lambda J(\beta) \tag{6}$$

where L is the Loss and J the penalty. λ is the Regularization Parameter.

- Loss should be piecewise quadratic w.r.t. λ(almost).
- Penalty should be piecewise linear w.r.t. λ. (L_1&L_∞)

Piecewise Linear Regularized Paths. Rosset & Zhu [6]

- Interesting as we can use many different Loss functions here.
- Provides a fast way to compute L_1 penalty.

Algorithm: (Intuitively, see Rosset et al. For more details)

Begin with $\beta = 0$

While $\max(|\nabla L(\beta)| > 0)$

- d_1 = find When another variable NOT in model becomes as big as the ones in the model.
- d_2 = find if the path hits 0.
- d_3 = find if it hits a knot (non differentiable point)

Find step length d = $\min(d_1, d_2, d_3)$

take step $\beta = \beta + d\lambda$

- if $d = d_1$ then add variable to model
- if $d = d_2$ then remove variable
- if $d = d_3$ handle knot event.

Calculate new direction.

Stop and Ask

Any Questions Yet?

Different Types of Losses

- We seen M.S.E.
- Lets now see SVM(Hinge) and Robust Regression(Huber) Loss.

Example: L2-SVM [2]

L2-SVM(Non-separable)

$$min_{\beta_0,\beta}\ \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n}\xi_i, \tag{7}$$

subject to, for each i, $y_i(\beta_0 + x_i^T\beta) \geq 1 - \xi_i, \xi_i \geq 0$ (cortes, vapnik '95).

L2-SVM - equivalent formulation (called hinge loss formulation)

$$min_{\beta_0,\beta}\sum_{i=1}^{n}[1 - Y_i(\beta_0 + \beta^T x_i)]_+ + \frac{\lambda}{2}||\beta||^2 \tag{8}$$

Second formulation looks like to the Penalty + Loss formulation for Regularization problem we saw.

Hinge Loss(Non-differentiable at 1): Linear Penalty if misclassified, 0 penalty if classified correctly.

Equivalence of the formulations 7 & 8

$$min_{\beta_0,\beta}\ \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n}\xi_i, \tag{9}$$

subject to, for each i, $y_i(\beta_0 + x_i^T\beta) \geq 1 - \xi_i, \xi_i \geq 0$

- move ξ to the other side
- $$\xi_i \geq 1 - y_i(\beta_0 + x_i^T\beta) \tag{10}$$
- replace in 7
- $$min_{\beta_0,\beta}\ \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n}[1 - y_i(\beta_0 + x_i^T\beta)]_+, \tag{11}$$
- We append $+$ sign to the Loss formulation to show that we consider only positive ξ
- Next we take care of the constants and we end up with hinge loss formulation 8

- Remember the SVM formulation says that we have to find 'optimal' hyperplane that maximize margin and reduce misclassification.
- Let's see what hinge loss($[1 - y_i(\beta_0 + x_i^T \beta)]_+$) does

Correctly Classified:

$$Y_i = +1, \qquad \beta_0 + x_i^T \beta \geq 1,$$
$$Hingeloss = 1 - (\geq 1) \leq \quad 0. \therefore \xi_i = 0$$
$$Y_i = -1, \qquad \beta_0 + x_i^T \beta \leq -1,$$
$$Hingeloss = 1 - (\geq 1) \leq \quad 0. \therefore \xi_i = 0$$

Misclassified:

$$Y_i = +1, \qquad \beta_0 + x_i^T \beta \leq 0,$$
$$Hingeloss = 1 + (\geq 0) \geq \quad 0. \therefore \xi_i = Non - negative\ loss$$
$$Y_i = -1, \qquad \beta_0 + x_i^T \beta \geq 0,$$
$$Hingeloss = 1 + (\geq 0) \geq \quad 0. \therefore \xi_i = Non - negative\ loss$$

L2-SVM - equivalent formulation (called hinge loss formulation)

$$min_{\beta_0,\beta} \sum_{i=1}^{n} [1 - Y_i(\beta_0 + \beta^T x_i)]_+ + \frac{\lambda}{2}||\beta||^2 \quad (12)$$

- Looks a lot like the Penalty + Loss formulation.
- Here β's magnitude is related to the margin maximization (due to the unique way in we frame the problem).

Stop and Ask

Any Questions Yet?

Example: L1-SVM [8]

L1-SVM - equivalent formulation

$$min_{\beta_0,\beta} \sum_{i=1}^{n} [1 - Y_i(\beta_0 + \beta^T x_i)]_+ + \frac{\lambda}{2}||\beta|| \tag{13}$$

Hinge Loss(Non-differentiable at 1):

- Linear Penalty if misclassified,
- 0 penalty if classified correctly.

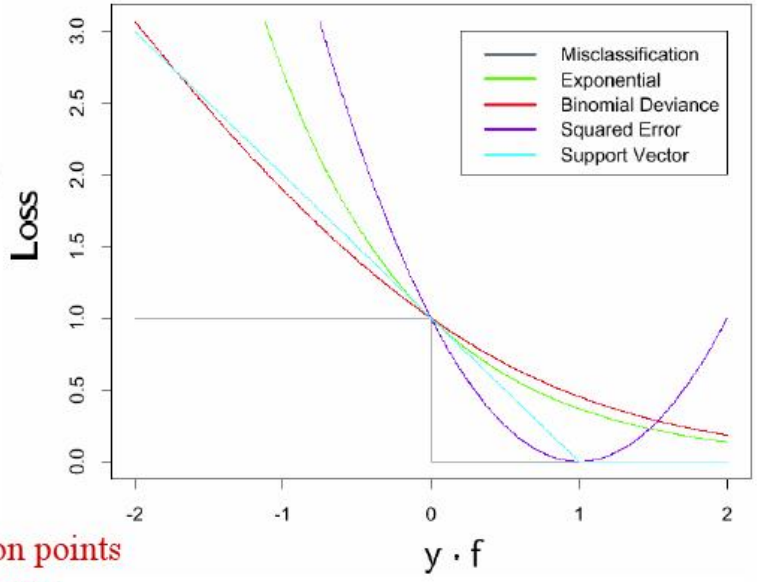
Non-smooth so we have a knot event.

Cost/Loss Functions(I)

For Classification $y = \pm 1,\ prediction = \hat{f}(x),\ Class\ prediction = \text{sgn}(\hat{f}(x))$

- Misclassification : $I(\text{sgn}(\hat{f}) \neq y)$
- Exponential : $\exp(-y\hat{f})$
- Binomial Deviance: $\log(1+\exp(-2y\hat{f}))$
- Squared Error : $(y-\hat{f}(x))^2$
- Support Vector : $(1-y\hat{f})\cdot I(y\hat{f}>1)$

Here, $I(x)=1$ *if* $x = TRUE$
$= 0$ *otherwise.*



Exponential error loss concentrates much more on points with large negative margins while **Binomial deviance** spreads the influence over all data. Hence, Binomial deviance is more robust in noise prone situations.

borrowed from Sethu Vijayakumar

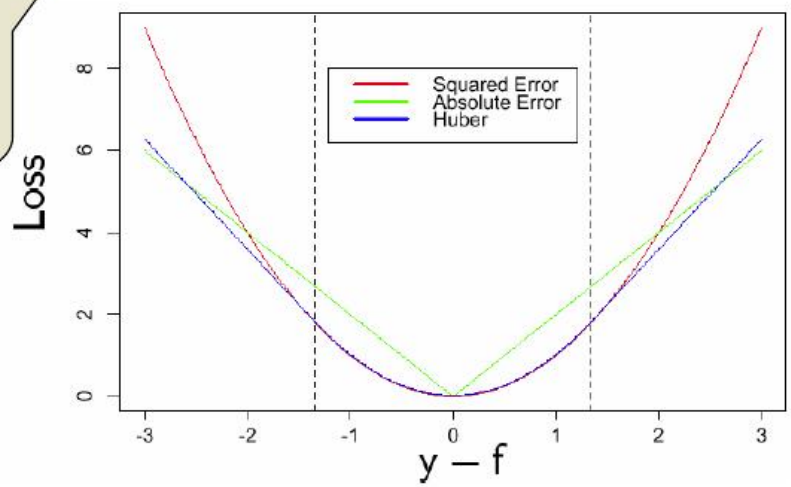
Cost/Loss Functions(II)

For Regression

- **Squared Error Loss :** $[y - \hat{f}(x)]^2$
- **Absolute Error Loss :** $| y - \hat{f}(x) |$
- **Huber Loss :** $[y - \hat{f}(x)]^2 \quad for \mid y - \hat{f}(x) \mid \leq \delta$
  $2\delta(\mid y - \hat{f}(x) \mid -\delta/2), \; otherwise$

} Cost Functions

Huber Loss combines the good properties of squared error loss near zero and absolute error loss when the error is large.



borrowed from Sethu Vijayakumar

Example: Huber Loss

- M.S.E has high values for Outliers,as it's quadratic. Thus not so robust to errors.
- Huber Loss is Robust. It gives quadratic error to points near margin and linear error to outliers.
- Has a knot event when transiting between linear and quadratic part.

In Conclusion

- We can get a Sparse solution with an L_1 penalty.
- We can solve with relatively good speed(compared to posing as an optimization problem) if we know how to formulate as a piecewise linear path.
- How to think about some ML problems in a Loss + Penalty formulation.

Now onto some hands-on examples

Thanks!

Other Stuff (Quiz)

1. Yet Another (Optimization way) penalty is L_∞ called as Dantzig selector. I talked about it and failed to mention that its sortof different(L_∞ vs. L_1) from LARS but gives similar result.
2. LARS is only related to L_1
3. Lambda parameter in Ridge regression alows Regularization(which means that it doesn't allow model to overfit). If you didn't tick this in the quiz its fine ;-)
4. Penalty term is optional, but Loss is required (what would you optimize on if there is no loss to optimize on?)

Quiz & solution

1. Which of the following(s) was/were the ML optimization formulation(s) we discussed today? (circle the correct one(s))
    1. penalty+regularizer
    2. loss+penalty✓
    3. loss✓ (it's fine if you didn't tick this up. I emphasized that + penalty was the way to go, so...)
    4. penalty
2. Which among the following norm penalties(L_3, L_2, $\underbrace{L_1}$) will give me a sparsest solution ? (circle one of them)
3. What use is ridge regression over just Least squares regression (without a penalty)? (circle the correct ones) - All of them are correct ;-)
    1. to avoid singular matrices in ill-posed problems.✓
    2. to make sure that β estimates don't have large variances.✓
    3. to get a stable solution (small pertubations won't make the β's jump a lot).✓
    4. to introduce regularization.✓
4. SVM's can be put in a Loss+Penalty formulation? (Yes or No) [just write Yes or No] Yes
5. What is the SVM loss termed as? (circle the correct one)
    1. exponential loss
    2. squared error loss
    3. hinge loss✓
    4. huber loss
6. LARS regression solves L_∞ penalty. (Yes or No)[just write Yes or No] No. LARS only for L_1 penalty

Appendix: References


Stephen P. Boyd and Lieven. Vandenberghe.
Convex optimization.
Cambridge University Press, Cambridge, UK; New York, 2004.


Corinna Cortes and Vladimir Vapnik.
Support-vector networks.
Machine Learning (Historical Archive), 20(3):273–297, 1995.


Donald W.Marquardt and Ronald D.Snee.
Ridge regression in practice.
The American Statistician, 29(1):3–20, feb 1975.


B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani.
Least angle regression, 2002.


Trevor Hastie, Jonathan Taylor, Robert Tibshirani, and Guenther Walther.
Forward stagewise regression and the monotone lasso.


Saharon Rosset and Ji Zhu.

Piecewise linear regularized solution paths.
Annals of Statistics, (35), 2007.

R. Tibshirani.
Regression shrinkage and selection via the lasso, 1994.

J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani.
1-norm support vector machines.
Technical report, Stanford University, 2003.