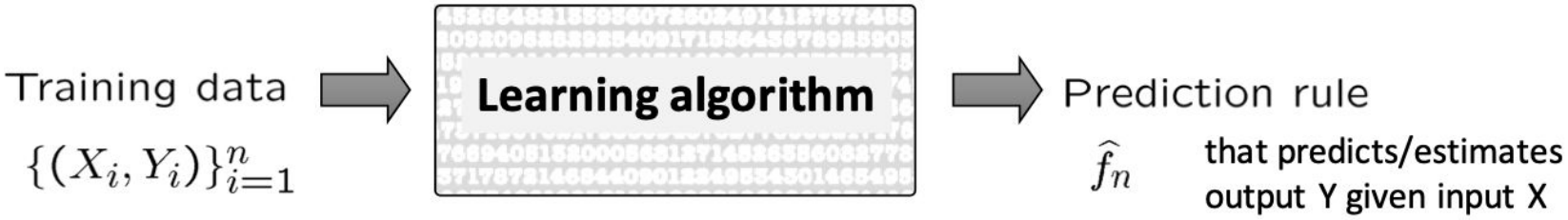
Modern Regression

GR 5205 / GU 4205
Section 3

Columbia University
Xiaofei Shi

Regression Task



Recap of linear regression:

Potential problems:

- collinearity
- too many non-zero but very small coefficients
- too slow

- Build your model:

  1) relationship:

  2) preference: choose w to minimize $J(\mathbf{w}) = \sum_i (y^i - \sum_j w_j \phi_j(x^i))^2$

- Estimate your model parameters:

  1) plugging in observed data to express your preference

  2) get parameters estimation for your model

- Understand your model $\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$

Choosing models

- It is always preferable to build more than one model with your data and choose the best among them.
- We cannot just use MSE, since the model with more variables will always give us a smaller MSE.
- We want a model that will predict the future observations well: we want a model with the smallest generalization error!

Generalization

- Denote the prediction as $\widehat{m}(x) = \widehat{\beta}_0 + \widehat{\beta}_1 x_1 + \cdots + \widehat{\beta}_1 x_p$
- Generalization error $G = \mathbb{E}[(Y - \widehat{m}(X))^2].$

  Test error $$T = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{m}(X_i))^2.$$

- We should use G instead of T when choosing the model.
- Relationship: the training error is always less than the generalization error

Why the training error always smaller than the generalization error?

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(Y_i' - \widehat{Y}_i)^2\right] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2\right] + \frac{2}{n}\sigma^2(p+1).$$

Optimism of the linear regression model

- more noise gives the model more opportunities to seem to fit well by capitalizing on chance.
- at any fixed level of noise, more data makes it harder to pretend the fit is better than it really is.
- every extra parameter is another control which can be adjusted to fit to the noise.
- Minimizing the in-sample MSE completely ignores the bias from optimism, so it is guaranteed to pick models which are too large and predict poorly out of sample.

Cross-validation: the best way to estimate generalization error

- Two main flavors: K-fold cross-validation and Leave-one-out cross-validation
- K-fold cross-validation
  - Randomly divide the data into K equally-sized parts, or “folds”.
  - For each folds:
    - Temporarily hold this fold as “testing set”
    - Label the other K-1 folds as “training set”
    - Estimate parameters of each candidate model on the “training set”
    - Calculate the MSE of each candidate model on the “testing set”
  - Average MSEs over different folds.
  - Pick the model with the lowest averaged MSE.

Mallow's C_p statistics

$$C_p = MSE + (\text{penalty})$$

- Leave-one-out cross-validation
- Score: $$LOOCV = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i^{(-i)})^2.$$

- Mallow's C_p statistic just substitutes in a feasible estimator of error term into the optimism

$$C_p = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2 + \frac{2\widehat{\sigma}^2}{n}(p+1).$$

Akaike Information Criterion (AIC)

$$AIC(S) \equiv L_S - \dim(S)$$

- If we specify the the linear-Gaussian models, then

$$L = -\frac{n}{2}(1 + \log 2\pi) - \frac{n}{2} \log MSE$$

$$\begin{aligned} \Delta AIC &\approx -\frac{n}{2}\frac{\Delta MSE}{\widehat{\sigma}^2} - (p_1 - p_2) \\ \frac{-2\widehat{\sigma}^2}{n}\Delta AIC &\approx \Delta MSE + \frac{2}{n}\widehat{\sigma^2}(p_1 - p_2) = \Delta C_p. \end{aligned}$$

Bayesian Information Criterion (BIC)

$$BIC(S) = L_S - \frac{\log n}{2}\dim(S).$$

- If the true model is among those BIC can select among, BIC will tend to pick the true model as sample size grows to infty.

Summary

- Cross-validation, AIC and Cp all have the same goal: try to find a model that predicts well. They tend to choose similar models.
- BIC is quite different and tends to choose smaller models.
- Cross-validation is very general and can be used in more settings than the others. There are theorems that say that cross-validation is very effective at estimating generalization error. These theorems make very few assumptions.

Variable Selection

- Suppose there are p covariates. For each variable, we can decide to keep it in the model or throw it away. This means that there are 2p possible models. In principle we could fit all 2p such models, estimate the prediction error of each one and choose the best. This presents a problem. There are too many models to consider.
- Then we need to introduce the stepwise method for model selection.

Forward Stepwise Regression

- Initial step: Consider all models with just one predictor, select the predictor that minimizes the criterion of interest.
  Equivalently, select the predictor that is most highly correlated with the response variable.
- Update rule: For each remaining step, consider adding one term to the subset selected at the previous step.
  Choose the term to minimize the criterion of interest. Equivalently, add the term with the highest partial correlation.
- Stopping rule: Stop when either all terms are in the model, or adding a term will only increase the selection criterion.

Backward Elimination

- Initial step: Fit the full model with all predictors included.
- At each next step, drop the term whose elimination leads to the greatest reduction to the selection criterion.
- Stop when either there are no terms left, or deleting any term would only increase the selection criterion.

Comparison

- There is no guarantee that forward selection and backward elimination will lead to the same model.
- There is no guarantee that either will find the “best” subset of predictors.
- In the end we use a combination of

  - selection criteria

  - computational tricks (search strategies) like forward selection and backward elimination

  - judgment, both statistical and subject matter.

Regularizer: ridge regression

- If $\Phi^T\Phi$ is not invertible, or its determinant is very small, the optimal w is not going to be stable
- n equations < p unknowns – underdetermined system of linear equations many feasible solutions

Need to impose extra constraints!

Regularizer: ridge regression

- If $\Phi^T\Phi$ is not invertible, or its determinant is very small, the optimal w is not going to be stable
- n equations < p unknowns – underdetermined system of linear equations many feasible solutions
- Adding in penalty term into loss function:

$$J(\beta) = \sum_i \left(y^i - \sum_j \beta_j \phi_j(x^i)\right)^2 + \lambda \sum_j \beta_j^2$$
$$= \|y - \Phi(x)\beta\|_2^2 + \lambda\|\beta\|_2^2$$

- Equivalent to a MAP optimization problem

$$\widehat{\beta} = (\Phi^\top(x)\Phi(x) + \lambda I)^{-1}\Phi^\top(x)y$$

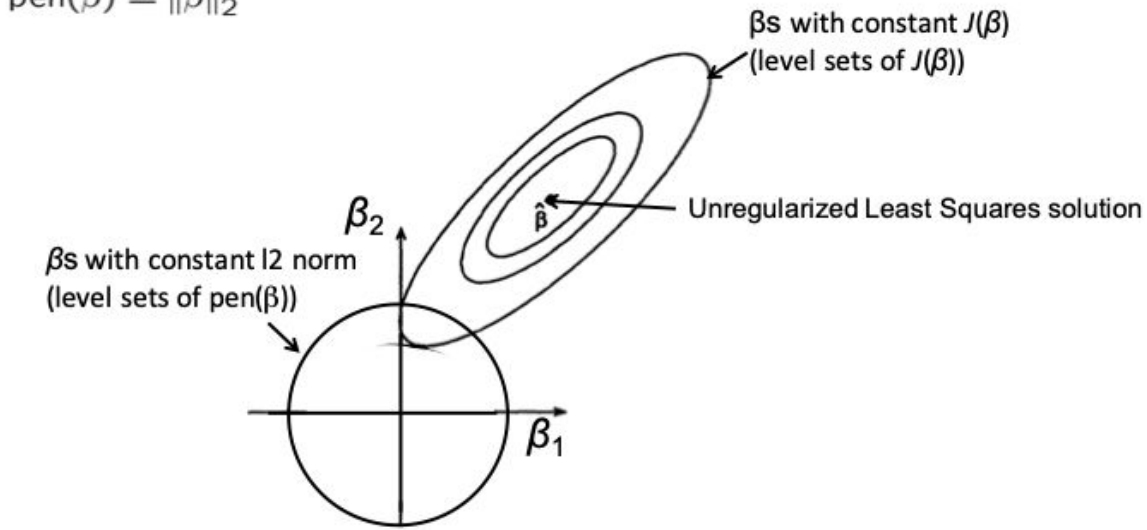
- Don't have to worry about invertibility anymore!

different norms of matrix and vectors

Regularizer: ridge regression

Ridge Regression:

$\text{pen}(\beta) = \|\beta\|_2^2$



Regularizer: lasso

- n equations < p unknowns – underdetermined system of linear equations many feasible solutions
- Sometimes our goal is to learn a ***sparse*** representation: select the most useful features!
- How to achieve?

Regularizer: lasso

- n equations < k unknowns – underdetermined system of linear equations many feasible solutions
- Sometimes our goal is to learn a ***sparse*** representation: select the most useful features!
- How to achieve?

$$J(\beta) = \|y - \Phi(x)\beta\|_2^2 + \lambda\|\beta\|_0$$

No closed form!
Hard to solve!

Regularizer: lasso

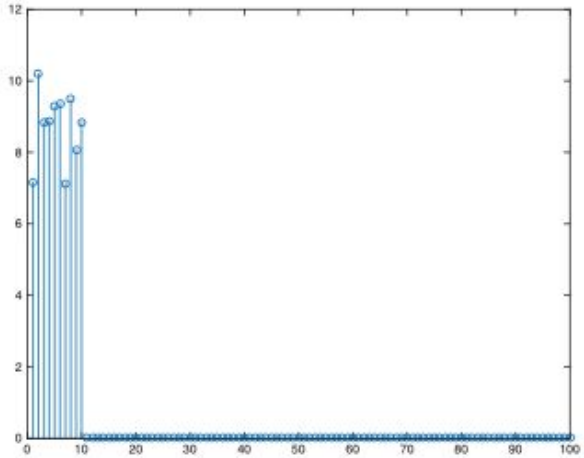
- n equations < k unknowns – underdetermined system of linear equations many feasible solutions
- Sometimes our goal is to learn a ***sparse*** representation: select the most useful features!
- How to achieve?

$$J(\beta) = \|y - \Phi(x)\beta\|_2^2 + \lambda\|\beta\|_1$$

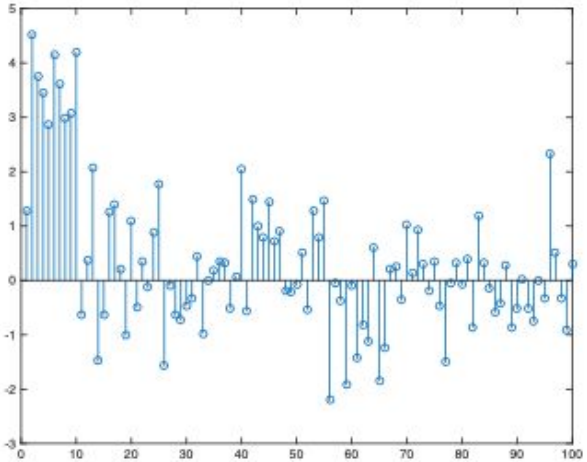
No closed form!
Getting easier!

Lasso or Ridge?

Lasso Coefficients



Ridge Coefficients

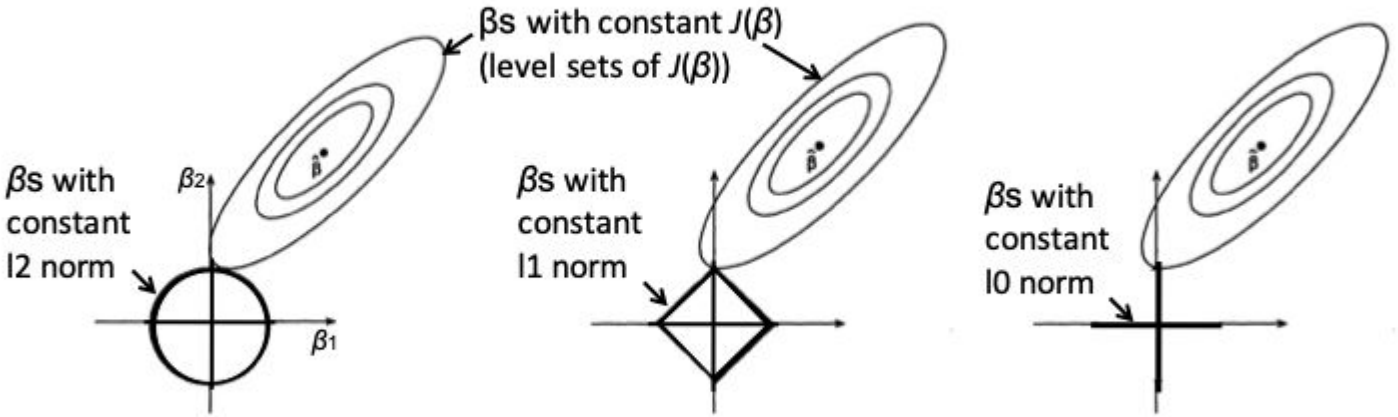


Lasso vs ridge

Ridge Regression:
$\text{pen}(\beta) = \|\beta\|_2^2$

Lasso:
$\text{pen}(\beta) = \|\beta\|_1$

Ideally l0 penalty, but optimization becomes non-convex



Lasso (l1 penalty) results in sparse solutions – vector with more zero coordinates
Good for high-dimensional problems – don't have to store all coordinates, interpretable solution!

References and further reading

- Christopher Bishop: *Pattern Recognition and Machine Learning*
- Kevin Murphy: *Machine Learning: A probabilistic perspective*
- Trevor Hastie, Robert Tibshirani, Jerome Friedman: *The Elements of Statistical Learning: Data Mining, Inference and Prediction*