MSA220/MVE440 Statistical Learning for Big Data

Lecture 7/8 - High-dimensional modeling part 1

Rebecka Jörnsten

Mathematical Sciences
University of Gothenburg and Chalmers University of Technology

Classification - feature selection

- Filter - select features based on some property
- Wrapper - build into classification algorithm
- Embedding - make selection part of the goal of the method

Classification - feature selection

Filter methods

- F-test or similar (Between/Within class variation) - what if many classes? imbalance?
- information theoretic: reduction in entropy (spread) when we condition on class label - generalization to mixed, discrete data - but see above
- Based on distances to observations with same label and average across other labels (ReliefF) - local query.
- Many variants.... careful how much they impact specific classifiers - they are univariate selectors and make assumption about relevance.
- Write your own or there are packages like FSelector (and GeneSelector in the Bioconductor environment).

Classification - feature selection

Filter methods

- Careful - most filtering methods work feature-by-feature, i.e. univariate approach
- Also - you may want to check that your top-ranked features aren't all separating one class from the others
- You can use one-against-all and use top-ranked features from all such comparisons - or do postprocessing on tests for all-pair comparisons
- Pro: very fast and simple and scales up easily.
- Con: filter is not connected to the learning process so may not pick features that work for the method you intend to use
- Tuning: Mainly filtering is used to reduce the dimension of the problem and this would be followed by some feature selection that is more advanced.
- ... but you can certainly use CV to pick the number of filtered features to use.

Classification - feature selection

Wrapper methods

- Incorporate selection into the classification method - e.g. by evaluating performance as function of selection.
- How search for set to evaluate? All-subset = expensive, Forward/backward = greedy, hybrid/random = needs many iterations
- RFE (recursive feature estimation) is a kind of wrapper - sometimes refers to backward selection and in other cases to a backward elimination based on variable importance.
- Implemented in caret for a few methods but not all. Backward selection is not a difficult wrapper to write yourselves though.

Classification - feature selection

Embedding

- Introduce a *penalty* on the number of features used - that's a hard selection problem...
- ... which we make practical by using sparsity constraints on feature-specific coefficients: L1-penalty (lasso, and variants thereof)!
- For Mini3 - the L1-penalized logistic regression can be investigated (package glmnet - also in caret). Also all the regularized discriminant methods we've looked at.
- First a brief introduction to L1-penalized modeling.

Intro - penalized regression

We want to fit a regression model to our data using least squares.

$$y = X\beta + \epsilon$$

- y is our $n \times 1$ vector of outcome data
- X is our $n \times p$ design matrix
- ϵ is our $n \times 1$ vector with additive errors.
- For convenience, assume centered and standardized data.

Is this OK?
Reality check: 5 basic assumptions, scatter plots,....
TRANSFORMATIONS! ID EXTREME OUTLIERS!

Regularized regression

When p is large or covariates in X are correlated, it is a tricky business to fit regression via OLS.
Why?

- $\min_\beta ||y - X\beta||^2$ has closed form solution
- $(X'X)^{-1}X'y$
- IF the inverse of $X'X$ exists.
- Not true if $p > n$. Inverse unstable if some covariates extremely correlated.

Regularized regression

What do we do?

- Reduce the number of covariates - prefiltering
- PCA of X and use only leading components.
- Partial least squares (more later)
- Regularized regression

Ridge regression

Regularization: To supress variance (due to instability of inverse of $X'X$), be willing to accept some bias!

- Ridge regression:
- $(X'X + \lambda I)^{-1}X'y$
- If $X'X = I$, this estimate $\beta_R = \beta_{OLS}/(1+\lambda)$ so biased but with lower variance
- If Xs are correlated, ridge regression *shrinkage* acts mostly on the directions with lower eigenvalues which correspond to the high variance estimates!
- See regression notes (MVE190/MSG500) for more on this.

Ridge regression

An alternative formulation of the ridge regression problem through penalized least squares.
We want to minimize

$$||y - X\beta||^2$$

$$\text{subject to } ||\beta||_2^2 \leq \tau$$

I.e., try to minimize least squares but don't let the average β get too big...

- Lagrangian formulation: $\min_\beta \frac{1}{2}||y - X\beta||^2 + \lambda||\beta||_2^2$
- Take derivatives with respect to β
- $-X'(y - X\beta) + \lambda\beta = 0$
- Solution $\beta_R = (X'X + \lambda I)^{-1}X'y$
- Choose λ to make sure condition τ holds or more commonly, choose λ via Cross-validation

Ridge regression

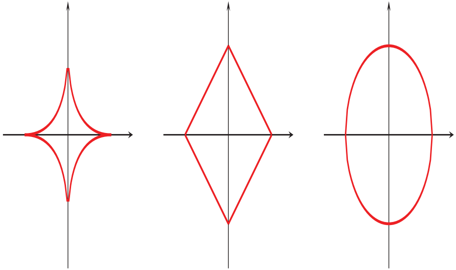
Pros and Cons?

- Pro: easy!
- Pro: can write other types of penalties here as well $\lambda\beta'\Omega\beta$ to penalize βs in a desired way
- Con: bias biggest for large coefficients
- Con: full model always returned since β_R may become very small but never exactly 0.

Lq-penalized regression

We can adress the growing bias and the lack of model interpretability using a different kind of penalty.

- L_0: $\min_\beta ||y - X\beta||^2 + \lambda \sum_{j=1}^{p} 1\{\beta_j \neq 0\}$
- L_q: $\min_\beta ||y - X\beta||^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q$
- L_1: $\min_\beta ||y - X\beta||^2 + \lambda \sum_{j=1}^{p} |\beta_j|$
- L_1: $\min_\beta ||y - X\beta||^2 + \lambda \sum_{j=1}^{p} \beta_j^2$



Fraction $q < 1$, $q = 1$ and $q = 2$

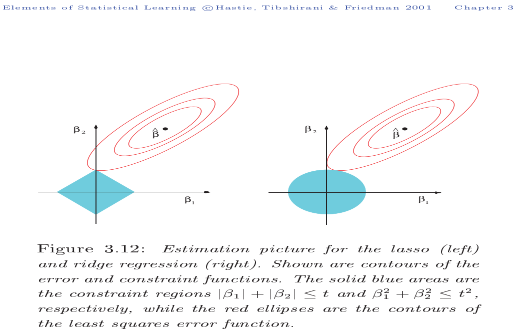
Lq-penalized regression

$q = 0$ is the penalty that corresponds to optimal model selection, we only count the number of variables included in the model.

Pro: no bias. Con: since the penalty is non-convex it is very difficult to work with.

$q = 1$ is the smallest q that provides a convex penalty AND has the nice property of performing selection.

Why?

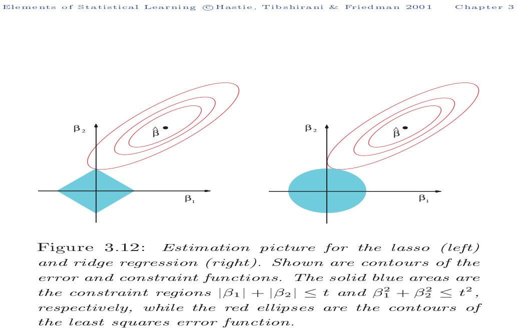


Elements of Statistical Learning ©Hastie, Tibshirani & Friedman 2001 Chapter 3

Figure 3.12: *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions $|\beta_1| + |\beta_2| \leq t$ and $\beta_1^2 + \beta_2^2 \leq t^2$, respectively, while the red ellipses are the contours of the least squares error function.*

Because the $L1$ penalty has "singularities" (points) this makes the selection of solutions at those points more likely.

We will see this by solving the problem mathematically too, but think of this as the penalty region extremes being the most likely to lead to a solution that is optimal for the loss (model fit).



Elements of Statistical Learning ©Hastie, Tibshirani & Friedman 2001 Chapter 3

Figure 3.12: *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions $|\beta_1| + |\beta_2| \leq t$ and $\beta_1^2 + \beta_2^2 \leq t^2$, respectively, while the red ellipses are the contours of the least squares error function.*

L1-penalized regression

$$\frac{1}{2}||y - X\beta||^2 + \lambda||\beta||_1$$

Consider first the special case $X'X = I$.

$$\frac{1}{2}y'y - y'X\beta + \frac{1}{2}\beta'\beta + \lambda||\beta||_1 = ***$$

Take derivatives with respect to β_j:

$$\frac{\partial ***}{\partial\beta_j} = -x_j'y + \beta_j + \lambda\nu_j$$

where

$$\nu_j = \begin{cases} sign(\beta_j) & if \quad \beta_j \neq 0 \\ \{\nu_j : |\nu_j| \leq 1\} & if \quad \beta_j = 0 \end{cases} \tag{1}$$

L1-penalized regression

$$\frac{\partial ***}{\partial \beta_j} = -x_j'y + \beta_j + \lambda \nu_j$$

where

$$\nu_j = \begin{cases} sign(\beta_j) & if \quad \beta_j \neq 0 \\ \{\nu_j : |\nu_j| \leq 1\} & if \quad \beta_j = 0 \end{cases} \tag{2}$$

So if $\beta_j > 0$, this is $\hat{\beta}_j = x_j'y - \lambda$ and if $\beta_j < 0$ this is $\hat{\beta}_j = x_j'y + \lambda$. There is a conflict between the assumed sign and the solution if $|x_j'y| < \lambda$. Note, $x_j'y = \hat{\beta}_j^{LS}$ for this special case $X'X = I$.
Solution:

$$\hat{\beta}_j = \begin{cases} \beta_j^{LS} - \lambda & if \quad \beta_j^{LS} > \lambda \\ \beta_j^{LS} + \lambda & if \quad \beta_j^{LS} < -\lambda \\ 0 & if \quad |\beta_j^{LS}| < \lambda \end{cases} \tag{3}$$

This is called the *Soft Thresholding operation*, ST and we write

$$\hat{\beta}_j = ST(x_j'y, \lambda)$$

L1-PENALIZED REGRESSION

What about the general case? We can't solve this with a closed-form expression. But there are tons of ways of solving this, numerically and iteratively.

We will look more into this in the upcoming lecture.

What does L1-penalties give us?

- Biased estimates → *adaptive lasso, SCAD* next lecture
- If $\lambda = o(n)$, then $\beta_{l1-pen} \to \beta_{true}$ as $n \to \infty$
- If $\lambda \propto n^{1/2}$ L1-pen has a non-zero probability of identifying the true model (model selection consistency) (Knight and Fu, 2000)
- BUT if many of the non-relevant variables are correlated with the relevant variables, L1-pen regression may fail to select the true model even if n is large.
- We need the *Irrepresentable condition* to hold

  $$|(X_1'X_1)^{-1}(X_2'X_2)| < 1 - \eta$$

  where X_1 are the irrelevant and X_2 the relevant variables. (Zhao and Yu, 2006)

Regularized Discriminant Analysis

We already talked about one particular kind of penalized DA: when we used the inverse of $\hat{\Sigma}_W + \lambda I$ to rotate/sphere our data.
Last lecture we saw that discriminant analysis could also be formulated as a regression problem which means you could do feature selection at the regression step via e.g. lasso.
This method, and variants on the same theme, is called *sparse discriminant analysis*. `sparseLDA` package

Regularized Discriminant Analysis

Several methods have been proposed for regularizing the within-covariance estimates.

- In QDA we can penalize each individual within-class covariance toward a common covariance (LDA)
- We can regularize the common within-class covariance toward a diagonal matrix (RDA)
- We can assume that the within-covariance matrix *is* diagonal (naive bayes)
- We can use a ridge-penalized estimate of the covariance matrix (PDA)

NB and Shrunken centroids

A special case of Naive Bayes is to replace the within-covariance estimate by its diagonal component.
This means we assume that features are independent.
In high-dimensional settings this tends to work quite well! The classifier now works on each variable at a time

$$k(i) = \arg\min_{l} \sum_{j=1}^{p} \frac{(x_{ij} - \mu_{lj})^2}{\sigma_l^2}$$

where $k(i)$ is the optimal class for observation i.
Tibshirani et al (2002) proposed we not use all the variables for classification.

- Shrink the class means (centroids) toward a common value (after standardizing by the within-class standard deviation)
- We can regularize the common within-class covariance toward a diagonal matrix (RDA)
- We can assume that the within-covariance matrix *is* diagonal (naive bayes)

Shrunken centroids

- Use a diagonal covariance estimate $diag(\Sigma + s_0^2 I)$ (where a small s_0 is used to avoid having really small standard deviations in the denominator later on)
- Compute for each variable j

$$t_{kj}^* = \frac{\hat{\mu}_{kj} - \hat{\mu}_j}{m_k(s_j + s_0)}$$

  where $\hat{\mu}_j$ is the overall mean for variable j, $s_j = \hat{\Sigma}_{jj}$ and $m_k = \sqrt{\frac{1}{n_k} + \frac{1}{n}}$
- Apply a soft-threshold to t_{kj}^*: $t_{kj} = ST(t_{kj}^*, \Delta)$
- Define $\mu_{kj}^s = \mu_j + m_k(s_j + s_0)t_{kj}$
- Use these *shrunken centroids* in your classifier!
- `pamr` package

SC-RDA

In 2005, Guo et al, proposed a combination of RDA and shrunken centroids

- We can use the SC method to shrink the centroids, either in the original data space
- or in the rotated data space!
- $\arg\min_k (x - \mu_k^s)' \hat{\Sigma}_R^{-1} (x - \mu_k^s)$
- `rda` package

Mini3

Choose one of the following topics via the doodle. On real and/or simulated data:

1. **Imbalanced data?**: Investigate the impact of imbalanced class sizes and try out some of the up/downsampling methods (e.g. SMOTE).
2. **Filter vs Method**: Filtering is easy to apply and to scale up - but is it favoring some methods over others? Try 2-3 filtering methods and 2-3 methods - do you see any combinations that are better than others and if so can you think of a reason why?

Mini3

3. **Wrapper - Selection accuracy**: How well do the wrappers do? On simulated data and real data - which features are picked? How much does the search type matter (backward, forward, random, exhaustive). The mlr package has many different search strategies (check makeFeatSelControl... for more info).
4. **Wrapper vs Method**: Which methods benefit most from feature reduction?

Mini3

5. **Embedding - selection accuracy**: On simulated data - check how well embedding methods work (glmnet for penalized methods, but you can also try other methods if you like). Sample size effect? Number of classes? etc.
6. **Many classes** How well do the wrappers/filters work if you have many classes in your data set? Two-class problems vs K-class problem. Is feature selection more difficult when you have many classes?
7. **Unrelated/Correlated features** Investigate the feature selection problem in the context of unrelated features and correlated features. Here, you can have correlated between relevant feature, between unrelated features, between related and unrelated features, etc.