Lecture 7: High-Dimensional Linear Regression

Francis J. DiTraglia

April 6, 2014

1 Review of Matrix Decompositions

1.1 The QR Decomposition

Any $n \times k$ matrix A with full column rank can be decomposed as $A = QR$, where R is an $k \times k$ upper triangular matrix and Q is an $n \times k$ matrix with orthonormal columns. The columns of A are *orthogonalized* in Q via the Gram-Schmidt process. Since Q has orthogonal columns, we have $Q'Q = I_k$. It is *not* in general true that $QQ' = I$, however. In the special case where A is square, $Q^{-1} = Q'$.

Note: The way we have defined things here is here is sometimes called the "thin" or "economical" form of the QR decomposition, e.g. `qr_econ` in Armadillo. In our "thin" version, Q is an $n \times k$ matrix with orthogonal columns. In the "thick" version, Q is an $n \times n$ *orthogonal* matrix. Let $A = QR$ be the "thick" version and $A = Q_1 R_1$ be the "thin" version. The connection between the two is as follows:

$$A = QR = Q \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1 R_1$$

Least-Squares via the QR Decomposition We can calculate the least squares estimator of β as follows

$$\begin{aligned}\widehat{\beta} &= (X'X)^{-1}X'y = \left[(QR)'(QR)\right]^{-1}(QR)'y \\ &= \left[R'Q'QR\right]^{-1}R'Q'y = (R'R)^{-1}R'Qy \\ &= R^{-1}(R')^{-1}R'Q'y = R^{-1}Q'y\end{aligned}$$

In other words, $\widehat{\beta}$ is the solution to $R\beta = Q'y$. While it may not be immediately apparent, this is a much easier system to solve that the normal equations $(X'X)\beta = X'y$. Because R is *upper triangular* we can solve $R\beta = Q'y$ extremely quickly. The product $Q'y$ is a vector, call it v, so the system is simply

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1,n-1} & r_{1k} \\ 0 & r_{22} & r_{23} & \cdots & r_{2,n-1} & r_{2k} \\ 0 & 0 & r_{33} & \cdots & r_{3,n-1} & r_{3k} \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & r_{k-1,k-1} & r_{k-1,k} \\ 0 & 0 & \cdots & 0 & 0 & r_k \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_{k-1} \\ \beta_k \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_{k-1} \\ v_k \end{bmatrix}$$

Hence, $\beta_k = v_k/r_k$ which we can substitute into $\beta_{k-1}r_{k-1,k-1} + \beta_k r_{k-1,k} = v_{k-1}$ to solve for β_{k-1}, and so on. This is called **back substitution**. We can use the same idea when a matrix is *lower triangular* only in reverse: this is called **forward substitution**.

To calculate the variance matrix $\sigma^2(X'X)^{-1}$ for the least-squares estimator, simply note from the derivation above that $(X'X)^{-1} = R^{-1}(R^{-1})'$. Inverting R, however, is easy: we simply apply back-substitution *repeatedly.* Let A be the inverse of R, $\mathbf{a}_j$ be the jth column of A, and $\mathbf{e}_j$ be the jth element of the $k \times k$ identity matrix, i.e. the jth standard basis vector. Inverting R is equivalent to solving $R\mathbf{a}_1 = \mathbf{e}_1$, followed by $R\mathbf{a}_2 = \mathbf{e}_2$, and so on all the way up to $R\mathbf{a}_k = \mathbf{e}_k$. In Armadillo, if you enclose a matrix in `trimatu()` or `trimatl()`, and then request the inverse, the library will carry out backward

or forward substitution, respectively.

Othogonal Projection Matrices and the QR Decomposition Consider a projection matrix $P_X = X(X'X)^{-1}X'$. Provided that X has full column rank, we have begin

$$P_X = QR(R'R)^{-1}R'Q' = QRR^{-1}(R')^{-1}R'Q' = QQ'$$

Recall that, in general, it is *not* true that $QQ' = I$ even though $Q'Q = I$ because we're using the *economical* QR decomposition in which Q has orthonormal columns but may not be a square matrix. Just to make this completely transparent, consider a very simple example:

$$X = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Then, we have

$$X'X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

but

$$XX' = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

It's important to keep the fact that $UU' \neq I$ in mind when using the QR decomposition for more complicated matrix calculations.

1.2 The Singular Value Decomposition

The Singular Value Decomposition (SVD) is probably the most elegant result in linear algebra. It's also an invaluable computational and theoretical tool in statistics and econometrics. I can only give a brief overview here, but I'd encourage you to learn more when you have time. Some excellent references are Strang (1993) and Kalman (2002).

The SVD Any $m \times n$ matrix A of arbitrary rank r can be decomposed according to

$$X = UDV' = (\text{orthogonal})(\text{diagonal})(\text{orthogonal})$$

- U is an $m \times m$ orthogonal matrix whose columns contain the eigenvectors of AA'
- V is an $n \times n$ orthogonal matrix whose columns contain the eigenvectors of $A'A$
- D is an $m \times n$ matrix whose first r main diagonal elements are the *singular values* $d_1, \ldots, d_r$. All other elements of D are zero.
- The singular values $d_1, \ldots, d_r$ are the positive eigenvalues of $A'A$ which are *identical* to the positive eigenvalues of AA'.

The Four Fundamental Subspaces It turns out that the SVD provides orthonormal bases for each of the so-called "fundamental subspaces" of a matrix A. In particular:

1. **column space**: first r columns of U
2. **row space**: first r columns of V
3. **null space**: last $n - r$ columns of V
4. **left null space**: last $m - r$ columns of U

SVD for Symmetric Matrices If A is symmetric then, by the spectral theorem, we can write $A = Q\Lambda Q'$ where Λ is a diagonal matrix containing the eigenvalues of A and Q is an orthonormal matrix whose columns are the corresponding eigenvectors. In this case, $U = V = Q$ and D is simply the absolute value of Λ (i.e. negative eigenvalues become positive singular values).

SVD for Positive Definite Matrices If A is not only symmetric but *positive definite*, then $A = Q\Lambda Q'$ is the *same decomposition* as $A = UDV'$: $U = V = Q$ and $\Lambda = D$.

The "Economical" SVD The number of singular values equals r, the rank of A, which is at most $\max\{m, n\}$. This means that some of the columns of U or V will be *irrelevant* since they will be multiplied by zeros in D. Accordingly, most linear algebra libraries provide an "economical" SVD that only calculate the columns of U and V that are multiplied by non-zero values in D. In Armadillo, for example, the command is `svd_econ`.

We can write the economical SVD in summation form as

$$A = \sum_{i=1}^{r} d_i \mathbf{u}_i \mathbf{v}_i'$$

where $r = \text{rank}(A)$ and the singular values d_i are arranged in order from largest to smallest. In matrix form, this is given by:

$$\underset{(n\times p)}{A} = \underset{(n\times r)}{U}\ \underset{(r\times r)}{D}\ \underset{(r\times p)}{V'}$$

In the economical SVD, U and V may no longer be square, so they are not orthogonal matrices but their *columns* are still orthonormal.

Approximation Property of SVD The Frobenius norm of a matrix A is given by

$$||A||_F = \sqrt{\sum_{i=1}^{m}\sum_{i=1}^{n} a_{ij}^2} = \sqrt{\text{trace}(A'A)}$$

Using this norm as a measure of "approximation error", it can be shown that the SVD provides the *best low rank approximation* to a matrix X.

Using the "economical" form of the SVD, we can write

$$X = \sum_{i=1}^{r} d_i \mathbf{u}_i \mathbf{v}_i'$$

where the index is i is defined such that the *largest* singular value comes first, followed by the second largest, and so on. This expression gives the rank-r matrix X as a *sum* of r rank-1 matrices. Now suppose that the rank of X is large and we wanted to *approximate* X using a matrix $\widehat{X}_L$ with rank $L < k$. If we measure the reconstruction error using the Frobenius norm, it can be shown that the *truncated SVD*

$$\widehat{X}_L = \sum_{i=1}^{L} d_i \mathbf{u}_i \mathbf{v}_i'$$

provides the best rank L approximation to X. In other words, $\widehat{X}_L$ is the arg min over all rank L matrices of the quantity $||X - \widehat{X}_L||_F$. It is also possible to provide bounds on the quality of the approximation, and thus choose an appropriate truncation.

2 Gauss-Markov, meet James-Stein

Consider the linear regression model

$$\mathbf{y} = X\beta + \boldsymbol{\epsilon}$$

In Econ 705 you learned that ordinary least squares (OLS) is the minimum variance unbiased linear estimator of β under the assumptions $E[\boldsymbol{\epsilon}|X] = \mathbf{0}$ and $Var(\epsilon|X) = \sigma^2 I$. When the second assumption fails, you learned that generalized least squares (GLS) provides a lower variance estimator than OLS. All of this is fine, as far as it goes, but there's an obvious objection: why are we restricting ourselves to unbiased estimators? Generically, we know that there is a bias-variance tradeoff. So what happens if we allow ourselves to consider biased estimators? Does some form of the Gauss-Markov Theorem still hold?

Admissibility

2.1 The James-Stein Estimator

2.2 The Positive-Part James-Stein Estimator

3 Ridge Regression

Ridge regression is a technique that was originally designed to address the problem of multicollinearity. When two or more predictors are very strongly correlated, OLS can become unstable. For example, if x_1 and x_2 are *nearly* linearly dependent, a large positive coefficient β_1 could effectively *cancel out* a large negative coefficient β_2. Ridge Regression attempts to solve this problem by *shrinking the estimated coefficients towards zero and towards each other.* This is accomplished by adding a squared L_2-norm "penalty" to the OLS objective function, yielding

$$\widehat{\beta}_{Ridge} = \arg\min_{\beta} (\mathbf{y} - \mathbf{1}_n\beta_0 - X\beta)'(\mathbf{y} - \mathbf{1}_n\beta_0 - X\beta) + \lambda\beta'\beta$$

where $\mathbf{1}_n$ is an $(n \times 1)$ vector of ones, β_0 denotes the regression intercept and $\beta = (\beta_1, \ldots, \beta_p)'$ the remaining coefficients. Note that we do *not* penalize the intercept in Ridge Regression. The easiest and most common way to handle this is simply to de-mean both X and $\mathbf{y}$ before proceeding. so that there is no

intercept and the problem becomes

$$\widehat{\beta}_{Ridge} = \arg\min_{\beta}(\mathbf{y} - X\beta)'(\mathbf{y} - X\beta) + \lambda\beta'\beta$$

Throughout these notes we will assume that everything has been de-meaned so there is no intercept.

3.1 Ridge is *Not* Scale Invariant

When we carry out OLS, if we re-scale a regressor $\mathbf{x}$, replacing it with $c\mathbf{x}$ where c is some nonzero constant, then the corresponding OLS coefficient estimate is scaled by $1/c$ to compensate. In other words, OLS is *scale invariant*. The same is not true of Ridge Regression, so it is common to convert the columns of the design matrix to the same units before proceeding. The usual way of handling this is simply to *standardize* each of the regressors.

3.2 Another Way to Express Ridge Regression

Data-dependent mapping.

3.3 Ridge as Bayesian Linear Regression

As you may recall from the first part of the semester, Bayesian models with informative priors automatically provide a form of shrinkage. Indeed, many frequentist shrinkage estimators can be expressed in Bayesian terms. Provided that we ignore the regression constant, the solution to Ridge Regression is *equivalent* to MAP (maximum a posteriori) estimation based on the following Bayesian regression model

$$\begin{aligned} y|\mathbf{x}, \beta, \sigma^2 &\sim N(\mathbf{x}'\beta|\sigma^2) \\ \beta &\sim N_p(\mathbf{0}, \tau^2 I_p) \end{aligned}$$

where σ^2 is assumed known and $\lambda = \sigma^2/\tau^2$. In other words, Ridge Regression gives the **posterior mode**. Since this model is conjugate, the posterior is normal. Thus, in addition to being the MAP estimator, the solution to Ridge Regression is also the posterior mean.

3.4 An Explicit Formula for Ridge Regression

The objective function is

$$\begin{aligned} Q_{ridge} &= (\mathbf{y} - X\beta)'(\mathbf{y} - X\beta) + \lambda\beta'\beta \\ &= \mathbf{y}'\mathbf{y} - \beta' X\mathbf{y} - \mathbf{y}'X\beta + \beta' X'X\beta + \lambda\beta' I_p \beta \\ &= \mathbf{y}'\mathbf{y} - 2\mathbf{y}'X\beta + \beta'(X'X + \lambda I_p)\beta \end{aligned}$$

Recall the following facts about matrix differentiation[1]

$$\begin{aligned} \frac{\partial(\mathbf{a}'\mathbf{x})}{\partial \mathbf{x}} &= \mathbf{a} \\ \frac{\partial(\mathbf{x}'A\mathbf{x})}{\partial \mathbf{x}} &= (A + A')\mathbf{x} \end{aligned}$$

Thus, we have

$$\frac{\partial}{\partial \beta} Q(\beta) = -2X'\mathbf{y} + \mathbf{2}(\mathbf{X}'\mathbf{X} + \lambda \mathbf{I_p})\beta$$

since $(X'X + \lambda I_p)$ is symmetric. Thus, the first order condition is

$$X'\mathbf{y} = (X'X + \lambda I_p)\beta$$

Hence,

$$\widehat{\beta}_{Ridge} = (X'X + \lambda I_p)^{-1} X'\mathbf{y}$$

So is $(X'X + \lambda I_p)$ guaranteed to be invertible? We need this to be the case for the solution of the Ridge Regression problem to be unique. In the following section, we'll provide an alternative way of analyzing the problem by turning

[1] See, for example, Harville (1997; Chapter 15).

it into something we're more familiar with: OLS.

3.5 Ridge Regression via OLS

From the first half of the semester, you may recall that Bayesian linear regression can be thought of as "plain-vanilla" OLS using a design matrix that has been *augemented* with "fake" observations that represent the prior. This turns out to be a very helpful way of looking at Ridge Regression. Define

$$\widetilde{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_p \end{bmatrix}, \qquad \widetilde{X} = \begin{bmatrix} X \\ \sqrt{\lambda} I_p \end{bmatrix}$$

The objective function for Ridge Rgression is *identical* to the OLS objective function for the augmented dataset, namely

$$\arg\min_{\beta} \left(\widetilde{\mathbf{y}} - \widetilde{X}\beta\right)' \left(\widetilde{\mathbf{y}} - \widetilde{X}\beta\right)$$

Which we can show as follows:

$$\begin{aligned} \left(\widetilde{\mathbf{y}} - \widetilde{X}\beta\right)' \left(\widetilde{\mathbf{y}} - \widetilde{X}\beta\right) &= \begin{bmatrix} (\mathbf{y} - X\beta)' & (-\sqrt{\lambda}\beta)' \end{bmatrix} \begin{bmatrix} (\mathbf{y} - X\beta) \\ -\sqrt{\lambda}\beta \end{bmatrix} \\ &= (\mathbf{y} - X\beta)'(\mathbf{y} - X\beta) + \lambda\beta'\beta \end{aligned}$$

3.6 Ridge is Always Unique

We know that the OLS estimator is only unique provided that the design matrix has full column rank. In constrast there is *always* a unique solution to the Ridge Regression problem, even when there are more regressors than observations. This follows *immediately* from the preceding: the columns of $\sqrt{\lambda} I_p$ are linearly independent, so the columns of the augmented data matrix $\widetilde{X}$ are *also* linearly independent, *regardless* of whether the same holds for the columns of X. Thus we can use Ridge Regression even in settings in which

there are more regressors than observations!

3.7 Efficient Calculations for Ridge Regression

Calculations When $p \gg n$

3.8 Predictive Bias and Variance of OLS and Ridge

Take the *economical* singular value decomposition of the $(n \times p)$ centered (and possibly standardized) design matrix X. We have

$$\underset{(n\times p)}{X} = \underset{(n\times r)}{U}\,\underset{(r\times r)}{D}\,\underset{(r\times p)}{V'}$$

where $r = \text{rank}(X)$ and thus

$$X'X = (UDV')'(UDV') = VDU'UDV' = VD^2V'$$

Provided that the columns of X are linearly indepdendent, $r = p$ and hence VD^2V' is the *eigen-decomposition* of $X'X$. Since X is centered, the sample covariance matrix of the regressors is $S = X'X/n$. Since it is simply a scalar multiple of $X'X$, the sample covariance matrix S has the *same eigenvectors* as $X'X$, namely the columns of V.

Continuing under the assumption that $r = p$ so that V is $(p \times p)$ and $V'V = V'V = I_p$, we have

$$\begin{aligned}
\widehat{\beta}_{Ridge} &= (X'X + \lambda I_p)^{-1} X'\mathbf{y} \\
&= \left(VD^2V' + \lambda I_p\right)(UDV')'\mathbf{y} \\
&= \left(VD^2V' + \lambda VV'\right) VDU'\mathbf{y} \\
&= V(D^2 + \lambda I_p)V'VDU'\mathbf{y} \\
&= V(D^2 + \lambda I_p)DU'\mathbf{y} \\
&= \sum_{i=1}^{p} \mathbf{v}
\end{aligned}$$

3.9 Choosing λ

Degrees of freedom, compare to OLS. AIC, BIC, cross-validation, k-fold.

4 Principal Components Regression

5 LASSO