MATH 680 Computation Intensive Statistics

Matrix Decomposition, PCA and Ridge Regularization

Yi Yang

McGill University

September 5, 2018

Flop Counts

- flop (floating-point operation): one addition, subtraction, multiplication, or division of two floating-point numbers
- to estimate complexity of an algorithm: express number of flops as a (polynomial) function of the problem dimensions, and simplify by keeping only the leading terms
- not an accurate predictor of computation time on modern computers
- useful as a rough estimate of complexity

Flop Counts

Vector-vector operations $(x, y \in \mathbb{R}^n)$

- Inner product $x^T y$: $2n - 1$ flops (or $2n$ if n is large)
- sum $x + y$, scalar multiplication αx: n flops

Matrix-vector product $y = Ax$ **with** $A \in \mathbb{R}^{n \times p}$

- $n(2p - 1)$ flops or $2np$ if p is large
- $2N$ if A is sparse with N nonzero elements
- $2k(n + p)$ if A is given as $A = UV^T$, $U \in \mathbb{R}^{n \times k}$, $V \in \mathbb{R}^{p \times k}$

Matrix-matrix product $C = AB$ **with** $A \in \mathbb{R}^{n \times k}, B \in \mathbb{R}^{k \times p}$

- $np(2k - 1)$ flops (or $2npk$ if k is large)
- less if A and/or B are sparse
- $(1/2)n(n + 1)(2k - 1) \approx n^2 k$ if $n = p$ and C symmetric

Notation

1. M^T : the transpose of M
2. $|A|$: the determinant of A
3. $\mathbb{S}^p = \{M \in \mathbb{R}^{p \times p} : M = M^T\}$: the set of symmetric matrices.
4. $\mathbb{S}^p_0 = \{M \in \mathbb{S}^p : \varphi_{\min}(M) \geq 0\}$: the set of symmetric and positive semi-definite matrices
5. $\mathbb{S}^p_+ = \{M \in \mathbb{S}^p : \varphi_{\min}(M) > 0\}$: the set of symmetric and positive definite matrices.

Introductory Example: Least Square Regression

- $X \in \mathbb{R}^{n \times p}$ be the nonrandom design matrix, where all entries in its first column equal 1
- $y = (y_1, \ldots, y_n)^T \in \mathbb{R}$ be the response vector
- Assume that

$$Y \sim N_n(X\beta_*, \sigma_*^2 I_n).$$

- The density for $N_n(\mu, \Sigma)$ and its logarithm evaluated at $x \in \mathbb{R}^n$ are

$$\phi(x; \mu, \Sigma) = (2\pi)^{-n/2} |\Sigma^{-1}|^{1/2} \exp\left\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right\},$$

$$\log \phi(x; \mu, \Sigma) = -\frac{n}{2}\log(2\pi) + \frac{1}{2}\log|\Sigma^{-1}| - \frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu).$$

Introductory Example: Least Square Regression

- The random log-likelihood function $\ell(\cdot,\cdot;Y) : \mathbb{R}^p \times \mathbb{R}_+ \to \mathbb{R}$ is defined by

$$\begin{aligned}
\ell(\beta,\sigma^2;Y,X) &= \log\phi(Y;X\beta,\sigma^2 I_n) \\
&= -\frac{n}{2}\log(2\pi) + \frac{1}{2}\log|\sigma^{-2}I_n| - \frac{1}{2}(Y-X\beta)^T\sigma^{-2}I_n(Y-X\beta) \\
&= -\frac{n}{2}\log(2\pi) + \frac{n}{2}\log\sigma^{-2} - \frac{1}{2}\sigma^{-2}(Y-X\beta)^T(Y-X\beta).
\end{aligned}$$

- The maximum likelihood estimator is

$$(\hat{\beta},\hat{\sigma}^2) = \operatorname*{arg\,min}_{(\beta,\sigma^2)\in\mathbb{R}^p\times\mathbb{R}_+} -\ell(\beta,\sigma^2;Y,X).$$

Introductory Example: Least Square Regression

- We call $-\ell$ the ***objective function***.
- (β, σ^2) the ***optimization variable***.
- $\mathbb{R}^p \times \mathbb{R}_+$ the ***feasible set***.
- To get $(\hat{\beta}, \hat{\sigma}^2)$, solve the two equations

$$\nabla_\beta - \ell(\beta, \sigma^2; Y, X) = 0$$

$$\nabla_{\sigma^2} - \ell(\beta, \sigma^2; Y, X) = 0$$

Introductory Example: Least Square Regression

We have that

$$(Y - X\beta)^T(Y - X\beta) = Y^TY - 2\beta^TX^TY + \beta^TX^TX\beta,$$

so

$$\begin{aligned}
\nabla_\beta - \ell(\beta, \sigma^2; Y, X) &= \frac{1}{2}\sigma^{-2}\nabla_\beta\{(Y - X\beta)^T(Y - X\beta)\} \\
&= \frac{1}{2}\sigma^{-2}\nabla_\beta\{Y^TY - 2\beta^TX^TY + \beta^TX^TX\beta\} \\
&= \frac{1}{2}\sigma^{-2}(-2X^TY + 2X^TX\beta) \\
&= \sigma^{-2}(-X^TY + X^TX\beta).
\end{aligned}$$

Introductory Example: Least Square Regression

- $\nabla_\beta - \ell(\beta, \sigma^2; Y, X) = 0$ is equivalent to $-X^T Y + X^T X\beta = 0$, so $\hat{\beta}$ solves $X^T X\hat{\beta} = X^T Y$.
- If $(X^T X)^{-1}$ exists, then $\hat{\beta} = (X^T X)^{-1} X^T Y$.
- Solving

$$\nabla_{\sigma^2} - \ell(\beta, \sigma^2; Y, X) = \frac{n}{2\sigma^2} - \frac{1}{2\sigma^4}(Y - X\beta)^T (Y - X\beta) = 0$$

we have that

$$\hat{\sigma}^2 = \frac{1}{n}(Y - X\beta)^T (Y - X\beta).$$

Ordinary Least Squares

- The OLS estimator of β_* is defined by

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} \|Y - X\beta\|_2^2, \tag{1}$$

  where here the arg min is a set of global minimizers:

- Let $f(\beta) = \|Y - X\beta\|^2$

$$f(\beta) = Y^T Y - 2\beta^T X^T Y + \beta^T X^T X\beta,$$

  and

$$\nabla f(\beta) = -2X^T Y + 2X^T X\beta.$$

- A minimizer $\hat{\beta}$ solves $\nabla f(\beta) = 0$

$$-X^T Y + X^T X\beta = 0.$$

QR Decomposition: solving $X^T X\hat{\beta} = X^T Y$

- Real matrix $A \in \mathbb{R}^{n\times p}$, $n > p$, has rank p.
- QR decomposition of A

$$A = QR$$

  - $Q \in \mathbb{R}^{n\times p}$ with $Q^T Q = I_p$
  - $R \in \mathbb{R}^{p\times p}$ is upper-triangular; $r_{ij} = 0$ when $i > j$, with non-zero diagonal entries.
  - Cost: $2np^2$ flops
- Since R is upper-triangular, $|R| = \prod_{j=1}^{p} r_{jj} \neq 0$ meaning that R is invertible.

QR Decomposition: solving $X^TX\hat{\beta} = X^TY$

To solve $X^TX\hat{\beta} = X^TY$

1. QR Decomposition: $2np^2$ flops

$$X = QR$$

$$R^TQ^TQR\hat{\beta} = R^TQ^TY$$

2. Backward substitution: (p^2 flops) since R is invertible (because R is upper-triangular and $|R| = \prod_{j=1}^{p} r_{jj} \neq 0$), we solve

$$R\hat{\beta} = Q^TY,$$

This costs $O(p^2)$.

QR Decomposition

```
set.seed(680)
n=10; p=5; sigma.star=1

## create the design matrix
Z=matrix( rnorm(n*(p-2)), nrow=n, ncol=(p-2))
v2=c( rep(1, n/2), rep(0, n/2) )
X=cbind(1, v2, Z)

## create the regression coefficient vector
beta.star=p^(-0.5) * rnorm(p)

## generate the responses
y=X%*%beta.star + sigma.star * rnorm(n)
```

QR Decomposition

- Slowest: closed-form solution

```
qr.solve(t(X)%*%X) %*% t(X)%*%y

##          [,1]
##     1.0343738
## v2 -0.7203269
##    -0.7991497
##    -0.1613902
##     0.3673047
```

QR Decomposition

- Slightly faster to use the `crossprod` function:

```
qr.solve(crossprod(X)) %*% crossprod(X,y)

##           [,1]
##      1.0343738
## v2 -0.7203269
##    -0.7991497
##    -0.1613902
##     0.3673047
```

QR Decomposition

- Faster to bypass the computation of $(X^T X)^{-1}$ and solve the linear system of equations $X^T X\beta = X^T Y$ directly:

```
qr.solve(crossprod(X), crossprod(X,y))

##           [,1]
##      1.0343738
## v2 -0.7203269
##     -0.7991497
##     -0.1613902
##      0.3673047

# equivalently
# qr.coef(qr(x=X), y=y) or lm.fit(x=X,y=y)$coef
# backsolve(qr.R(out), crossprod(qr.Q(out), y))
```

Cholesky Decomposition

- Suppose that $A \in \mathbb{S}^p_+$.
- A can be factored as $A = LL^T$,
  - $L \in \mathbb{R}^{p \times p}$ is lower-triangular: $l_{ij} = 0$ when $i < j$, with positive diagonal entries. Cholesky decomposition of A.
- Since L is lower-triangular, $|L| = \prod_{j=1}^{p} l_{jj} > 0$ so L is invertible.
- Cost: $(1/3)p^3$ flops

$X \in \mathbb{R}^{n \times p}$ with rank $p \implies X^T X \in \mathbb{S}^p_+$:

- For any $u \in \mathbb{R}^p$ for which $u \neq 0$, we have that $u^T X^T X u = (Xu)^T (Xu) = \|Xu\|^2 > 0$ because X has rank p.

Cholesky Decomposition

To solve $X^T X\beta = X^T Y$,

1. Cholesky factorization: $(1/3)p^3$ flops
$$X^T X = LL^T$$
2. Forward substitution : p^2 flops
$$Lz = X^T Y,$$
3. Backward substitution: p^2 flops
$$L^T \beta = z$$

Cholesky Decomposition

```
U=chol(crossprod(X))
z=forwardsolve(t(U), crossprod(X,y))
beta.hat.chol=backsolve(U, z)
```

Eigen Decomposition

- Let $A \in \mathbb{S}^p$ be a square matrix
- we can write

$$A = Q\Lambda Q^{-1}$$

  where $Q \in \mathbb{R}^{p\times p}$ is the matrix for **eigenvectors** and $\Lambda \in \mathbb{R}^{p\times p}$ is diagonal. The diagonal elements of Λ are the **eigenvalues** and are ordered $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_p$.
- As a special case, for every $p \times p$ real symmetric matrix, the eigenvalues are **real** and the eigenvectors can be chosen to be **orthogonal** (hence $Q^TQ = I_p$ to each other:

$$A = Q\Lambda Q^T$$

- Cost: $O(p^3)$ flops

Eigen Decomposition

- Eigen-decomposition allows for much easier computation of **power series of matrices**.
- Suppose that $A \in \mathbb{S}_0^p$ with eigen-decomposition $A = Q\Lambda Q^T$, then $\lambda_1, \ldots, \lambda_p \geq 0$. We have

$$A^x = Q\Lambda^x Q^T$$

where $\Lambda^x = \mathrm{diag}(\lambda_1^x, \ldots, \lambda_p^x)$ is a diagonal matrix with (j,j)-th entry λ_j^x and x is any real number.

Eigen Decomposition

- For example, $A^{1/2} = Q\text{diag}(\lambda_1^{1/2}, \ldots, \lambda_p^{1/2})Q^T$ so

$$\begin{aligned}
A^{1/2}A^{1/2} &= Q\text{diag}(\lambda_1^{1/2}, \ldots, \lambda_p^{1/2})Q^T Q\text{diag}(\lambda_1^{1/2}, \ldots, \lambda_p^{1/2})Q^T \\
&= Q\text{diag}(\lambda_1^{1/2}, \ldots, \lambda_p^{1/2})\text{diag}(\lambda_1^{1/2}, \ldots, \lambda_p^{1/2})Q^T \\
&= Q\text{diag}(\lambda_1, \ldots, \lambda_p)Q^T \\
&= A.
\end{aligned}$$

- If matrix A can be eigen-decomposed and if none of its eigenvalues are zero, then A is non-singular and its inverse is given by

$$A^{-1} = Q\Lambda^{-1}Q^T.$$

Eigen Decomposition

Generating an iid sample of size n from $N_p(\mu, \Sigma)$

1. Suppose that $\Sigma \in \mathbb{S}_0^p$. Z is an $n \times p$ random matrix, with all entries i.i.d $N(0, 1)$. Then $X = 1_n\mu^T + Z\Sigma^{1/2}$ has rows that are i.i.d $N_p(\mu, \Sigma)$.
   - The ith row vector of X is $X_i = \mu + \Sigma^{1/2}Z_i$
   - $E(X_i) = \mu + \Sigma^{1/2}E(Z_i) = \mu$
   - $var(X_i) = \Sigma^{1/2}var(Z_i)\Sigma^{1/2} = \Sigma^{1/2}I_p\Sigma^{1/2} = \Sigma$
2. Suppose that $\Sigma \in \mathbb{S}_+^p$ with Cholesky decomposition $\Sigma = LL^T$. Z is an $n \times p$ random matrix, with all entries i.i.d $N(0, 1)$. Then $X = 1_n\mu^T + ZL^T$ has rows that are i.i.d $N_p(\mu, \Sigma)$.
   - The ith row vector of X is $X_i = \mu + LZ_i$
   - $E(X_i) = \mu + LE(Z_i) = \mu$
   - $var(X_i) = Lvar(Z_i)L^T = LI_pL^T = \Sigma$

Full SVD

Singular Value Decomposition:

- Let A be an $n \times p$ real matrix $(n > p)$
- A can be factored as

$$A = UDV^T$$

  - $U \in \mathbb{R}^{n \times n}$ has orthogonal columns $U^T U = I_n$
  - $V \in \mathbb{R}^{p \times p}$ has orthogonal columns $V^T V = I_p$
  - $D \in \mathbb{R}^{n \times p}$ has **positive singular values** only along the diagonal.
  - The number of positive diagonal entries in D is the rank of A

Full SVD

Full SVD

Calculating SVD of X ($\text{rank}(X) = q$) is related to finding the eigenvalues and eigenvectors of XX^T and X^TX.

- The eigenvectors of X^TX make up the columns of V
  - $X = UDV^T$, $X^TX = VD^TU^TUDV^T = VD^TDV^T$
- The eigenvectors of XX^T make up the columns of U
- $d_1, \ldots, d_q$ in D are square roots of eigenvalues from XX^T or X^TX.

Reduced SVD

A real matrix $A \in \mathbb{R}^{n \times p}$ with $q = \text{rank}(A) = \min(n, p)$

$$A = UDV^T$$

where $U \in \mathbb{R}^{n \times q}$ and $V \in \mathbb{R}^{p \times q}$ has **orthogonal** columns.
$D \in \mathbb{R}^{q \times q}$ diagonal square matrix with **positive** entries $d_{11}, \ldots, d_{qq}$

- if $n \geq p$, then $q = p$. Extend U to $\hat{U} = \left[U, \tilde{U}\right] \in \mathbb{R}^{n \times n}$.

$$A = UDV^T = \underbrace{\left[U_{n \times q}, \tilde{U}_{n \times (n-q)}\right]}_{\hat{U}_{n \times n}} \underbrace{\begin{bmatrix} D_{q \times q} \\ 0_{(n-q) \times q} \end{bmatrix}}_{\hat{D}_{n \times q}} V^T_{q \times p} = \hat{U}\hat{D}V^T$$

- if $n < p$, then $q = n$. Extend V to $\hat{V} = \left[V, \tilde{V}\right] \in \mathbb{R}^{p \times p}$.

$$A = UDV^T = U_{n \times q} \underbrace{\begin{bmatrix} D_{q \times q} & 0_{q \times (p-q)} \end{bmatrix}}_{\hat{D}_{n \times p}} \underbrace{\begin{bmatrix} V_{p \times q} & \tilde{V}_{p \times (p-q)} \end{bmatrix}^T}_{\hat{V}^T_{p \times p}} = U\hat{D}\hat{V}^T$$

- This forces $D \in \mathbb{R}^{q \times q}$ to be extended to an $\mathbb{R}^{n \times p}$ matrix $\hat{D}$.

Ridge Regression

The ridge penalized least squares estimator of $\hat{\beta}^{(\lambda)}$ is defined by

$$\hat{\beta}^{(\lambda)} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \left\{ \|Y - X\beta\|^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\} \tag{2}$$

$$= \underset{\beta \in \mathbb{R}^p}{\arg\min} \|Y - X\beta\|^2 + \lambda \|\beta\|_2^2, \tag{3}$$

we solve $\nabla f(\beta^{(\lambda)}) = 0$ for $\beta^{(\lambda)}$, which is

$$\begin{aligned} -2X^TY + 2X^TX\beta^{(\lambda)} + 2\lambda\beta^{(\lambda)} &= 0 \\ (X^TX + \lambda I_p)\beta^{(\lambda)} &= X^TY \qquad (4) \\ \beta^{(\lambda)} &= (X^TX + \lambda I_p)^{-1}X^TY, \qquad (5) \end{aligned}$$

Ridge Regression

- Recomputing $\beta^{(\lambda)}$ for a different value of λ is **computationally expensive** when p is large and inefficient when compute $\beta^{(\lambda)}$ for multiple values of λ.
- We derive **a fast way** to compute $\beta^{(\lambda)}$.
- Suppose $\mathrm{rank}(X) = q = \min(n-1, p))$. By $X = UDV^T$, where $U = \mathbb{R}^{n\times n}$, $V \in \mathbb{R}^{p\times p}$ and $D = \mathbb{R}^{n\times p}$

$$X^T X = VD^T U^T UDV^T = VD^T DV^T$$

and

$$X^T = VD^T U^T.$$

Ridge Regression

So write (4) as

$$(VD^TDV^T + \lambda I_p)\beta^{(\lambda)} = VD^TU^TY$$

and replacing I_p with VV^T gives

$$V(D^TD + \lambda I_p)V^T\beta^{(\lambda)} = VD^TU^TY.$$

$V(D^TD + \lambda I_p)V^T$ is the **eigen decomposition** of $X^TX + \lambda I_p$, which is in $\mathbb{S}^p_+$ if $\lambda > 0$ or $X^TX \in \mathbb{S}^p_+$ (so it's **invertible**)

$$\begin{aligned} \beta^{(\lambda)} &= V(D^TD + \lambda I_p)^{-1}V^TVD^TU^TY \\ &= V(D^TD + \lambda I_p)^{-1}D^TU^TY \\ &= VMU^TY \end{aligned}$$

where the matrix $M = (D^TD + \lambda I_p)^{-1}D^T \in \mathbb{R}^{p\times n}$ is **diagonal** where $m_j = d_j/(d_j^2 + \lambda)$ for $j = 1, \ldots, q$.

Ridge Regression

- We can avoid multiplication by zero using the reduced SVD.
- Suppose $\operatorname{rank}(X) = q = \min(n-1, p))$.
- Decompose $X = UDV^T$, where $U \in \mathbb{R}^{n\times q}$, $V \in \mathbb{R}^{p\times q}$ and $D \in \mathbb{R}^{q\times q}$. One can prove that

$$\beta^{(\lambda)} = VMU^TY$$

where the square matrix $M \in \mathbb{R}^{q\times q}$ is diagonal with the entries $m_j = d_j/(d_j^2 + \lambda)$, for $j = 1, \ldots, q$. We see that

$$\lim_{\lambda\to 0^+} \beta^{(\lambda)} = VD^{-1}U^TY = X^-Y = \hat{\beta}^{OLS}.$$

Relationship between Ridge and PCA

- **OLS:** Write the least squares fitted vector as

$$
\begin{aligned}
X\hat{\beta}^{OLS} &= X(X^TX)^{-1}X^TY \\
&= UDV^T(VDU^TUDV^T)^{-1}VDU^TY \\
&= UU^TY
\end{aligned}
$$

  U^TY are the coordinates of Y wrt the basis U.

- **Ridge:** **greater shrinkage** is applied to **the coordinates of basis vectors with smaller** d_j^2.

$$
\begin{aligned}
X\hat{\beta}^{ridge} &= X(X^TX+\lambda I)^{-1}X^TY \\
&= UD(D^2+\lambda I)^{-1}DU^TY \\
&= \sum_{j=1}^{p} \mathbf{u}_j \frac{d_j^2}{d_j^2+\lambda} \mathbf{u}_j{}^T Y \\
&= UHU^TY
\end{aligned}
$$

  where $H = D(D^2+\lambda I)^{-1}D$ is diagonal with $h_j = d_j^2/(d_j^2+\lambda)$, for $j = 1, \ldots, q$.

Now let's understand d_j^2

- For example, the ξ_1 is the **first principal component** (PC) of X,

$$XV = DU$$

$$\xi_1 = X\mathbf{v}_1 = d_{11}\mathbf{u}_1$$

- Hence $\mathbf{u}_1$ is the **normalized first PC**

$$\begin{aligned}
Var(\xi_1) &= Var(X\mathbf{v}_1) = \mathbf{v}_1{}^T Var(X)\mathbf{v}_1 \\
&= \mathbf{v}_1{}^T X^T X\mathbf{v}_1/n \\
&= \mathbf{v}_1 VD^T U^T UDV^T\mathbf{v}_1/n \\
&= \mathbf{v}_1{}^T VD^T DV^T\mathbf{v}_1/n \\
&= \frac{d_1^2}{n}
\end{aligned}$$

- Ridge **shrinks the normalized PCs** $\mathbf{u}_j$ of these directions with **smallest variance the most**.