Lecture 9

Least Squares, QR and SVD

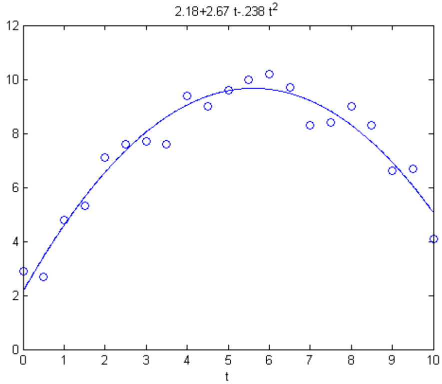
T. Gambill

Department of Computer Science
University of Illinois at Urbana-Champaign

March 15, 2011

Example 1: Finding a curve that best fits the data

Suppose we are given the data $\{(t_1, y_1), ..., (t_{21}, y_{21})\}$ (circles) and we want to find a parabolic curve that *best fits* the data.



Example 1: Finding a curve that best fits the data

We are looking for a curve of the form,

$$f(t, \mathbf{x}) = x_1 + x_2 t + x_3 t^2$$

so that

$$A * \mathbf{x} = \begin{bmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ \vdots & \vdots & \vdots \\ 1 & t_{21} & t_{21}^2 \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \approx \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{21} \end{bmatrix} = \mathbf{b}$$

The matrix A has the form of a **Vandermonde** matrix.

Example 2: Reducing Measurement Error

Suppose a surveyor determined the heights of three hills above some reference point as $x_1 = 1237$ft., $x_2 = 1941$ft. and $x_3 = 2417$ft. and to confirm these measurements the surveyor climbs to the top of the first hill and measures the height of the second hill above the first to be, $x_2 = x_1 + 711$ and the third above the first to be $x_3 = x_1 + 1177$. Similarly, the surveyor climbs the second hill and measures the height of the third above the second to be $x_3 = x_2 + 475$. (M. Heath) These equations can be written in matrix form as,

$$A * x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \approx \begin{bmatrix} 1237 \\ 1941 \\ 2417 \\ 711 \\ 1177 \\ 475 \end{bmatrix} = b$$

Systems with more equations than unknowns are called **overdetermined**
The system above, $Ax = b$ is an over-determined linear system. What values should the surveyor give for the heights of the hills?

Overdetermined Systems

If A is an $m \times n$ matrix, then in general, an $m \times 1$ vector b may not lie in the column space of A. Hence $Ax = b$ may not have an exact solution.

Definition

The **residual** vector is

$$r = b - Ax.$$

The **least squares** solution is given by minimizing the square of the residual in the 2-norm.

Normal equations

Writing $r = (b - Ax)$ and substituting, we want to find an x that minimizes the following function

$$\phi(x) = ||r||_2^2 = r^T r = (b - Ax)^T (b - Ax) = b^T b - 2x^T A^T b + x^T A^T A x$$

From calculus we know that the minimizer occurs where $\nabla\phi(x) = 0$.

The derivative is given by

$$\nabla\phi(x) = -2A^T b + 2A^T A x = 0$$

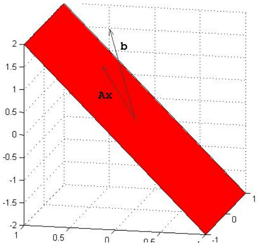
Definition

The system of **normal equations** is given by

$$A^T A x = A^T b.$$

The normal equations has a unique solution if $rank(A) = n$.

Normal equations, a Geometric View



If the vector b is not in the span (the set of all linear combinations of vectors) of the columns of A then in order to find the minimum distance from b to the span of the columns of A we need to find x^* such that $r = (b - Ax^*)$ is orthogonal to Ax for any x.

$$< r, Ax > = < b - Ax^*, Ax > = 0$$

Normal equations, a Geometric View

$$< r, Ax > = < b - Ax^*, Ax > = 0$$

or

$$< A^T(b - Ax^*), x > = 0$$

so that with $x = e_i$ the column vectors of the identity matrix we have,

$$(A^T(b - Ax^*))^T e_i = 0 \text{ for } i = 1, ...n$$

and thus

$$A^T b = A^T A x^*$$

Solving normal equations

Since the normal equations forms a symmetric positive definite system (assuming $rank(A) = n$), we can solve by computing the Cholesky factorization

$$A^T A = LL^T$$

and solving $Ly = A^T b$ and $L^T x = y$.

Consider

$$A = \begin{bmatrix} 1 & 1 \\ \epsilon & 0 \\ 0 & \epsilon \end{bmatrix}$$

where $0 < \epsilon < \sqrt{\epsilon_{mach}}$. The normal equations for this system is given by

$$A^T A = \begin{bmatrix} 1+\epsilon^2 & 1 \\ 1 & 1+\epsilon^2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Normal equations: conditioning

The normal equations tend to worsen the condition of the matrix.
Since we defined the condition number for a square matrix only we will have to extend this definition for $A_{m\times n}$.

Definition

Let $A_{m\times n}$ have $rank(A) = n$. Then we define the pseudo-inverse A^+ of A as
$A^+ = (A^TA)^{-1}A^T$
and we define the condition number of A as,
$cond(A) = ||A||_2||A^+||_2$

Theorem

$$cond(A^TA) = (cond(A))^2$$

Normal equations: Python conditioning example

```
>>> A = scipy.rand(10,10)
>>> np.linalg.cond(A)
162.83042743389382
>>> np.linalg.cond(np.dot(A.T,A))
26513.748098298413
```

How can we solve the least squares problem without squaring the condition of the matrix?

Other approaches

- QR factorization.
  - For $A \in \mathbb{R}^{m \times n}$, factor $A = QR$ where
    - ★ Q is an $m \times m$ orthogonal matrix
    - ★ R is an $m \times n$ upper triangular matrix (since R is an $m \times n$ upper triangular matrix we can write $R = \begin{bmatrix} R' \\ 0 \end{bmatrix}$ where R' is $n \times n$ upper triangular and 0 is the $(m-n) \times n$ matrix of zeros)
- SVD - singular value decomposition
  - For $A \in \mathbb{R}^{m \times n}$, factor $A = USV^T$ where
    - ★ U is an $m \times m$ orthogonal matrix
    - ★ V is an $n \times n$ orthogonal matrix
    - ★ S is an $m \times n$ diagonal matrix whose elements are the singular values.

Orthogonal matrices

Definition

A matrix Q is orthogonal if

$$Q^T Q = QQ^T = I$$

Orthogonal matrices preserve the Euclidean norm of any vector v,

$$||Qv||_2^2 = (Qv)^T (Qv) = v^T Q^T Qv = v^T v = ||v||_2^2.$$

Using QR factorization for least squares

Now that we know orthogonal matrices preserve the euclidean norm, we can apply orthogonal matrices to the residual vector without changing the norm of the residual. Note that A is $m\ x\ n$, Q is $m\ x\ m$, R' is $n\ x\ n$, x is $n\ x\ 1$ and b is $m\ x\ 1$.

$$\|r\|_2^2 = \|b - Ax\|_2^2 = \left\| b - Q \begin{bmatrix} R' \\ 0 \end{bmatrix} x \right\|_2^2 = \left\| Q^T b - Q^T Q \begin{bmatrix} R' \\ 0 \end{bmatrix} x \right\|_2^2 = \left\| Q^T b - \begin{bmatrix} R' \\ 0 \end{bmatrix} x \right\|_2^2$$

If $Q^T b = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$ where c_1 is an $n\ x\ 1$ vector then

$$\left\| Q^T b - \begin{bmatrix} R' \\ 0 \end{bmatrix} x \right\|_2^2 = \left\| \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} - \begin{bmatrix} R'x \\ 0 \end{bmatrix} \right\|_2^2 = \left\| \begin{bmatrix} c_1 - R'x \\ c_2 \end{bmatrix} \right\|_2^2 = \|c_1 - R'x\|_2^2 + \|c_2\|_2^2$$

Hence the least squares solution is given by solving $R'x = c_1$. We can solve $R'x = c_1$ using back substitution and the residual is $\|r\|_2 = \|c_2\|_2$.

Gram-Schmidt orthogonalization

One way to obtain the QR factorization of a matrix $A_{m\times n}$ ($rank(A) = n$) is by Gram-Schmidt orthogonalization.

For each column of A, subtract out its component in the other columns.

For the simple case of 2 vectors $\{a_1, a_2\}$, first normalize a_1 and obtain

$$q_1 = \frac{a_1}{||a_1||}.$$

Now subtract from a_2 the components from q_1. This is given by

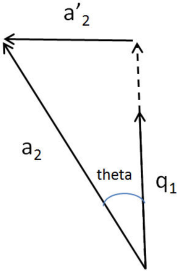
$$a_2' = a_2 - < q_1, a_2 > q_1 = a_2 - (q_1^T a_2) q_1.$$

Normalizing a_2' we have

$$q_2 = \frac{a_2'}{||a_2'||}$$

Repeating this idea for n columns gives us Gram-Schmidt orthogonalization.

Gram-Schmidt orthogonalization



$a_2 = a'_2 + c * q_1$

implies

$a'_2 = a_2 - c * q_1$

where c is determined by

$< q_1, a_2 > = || q_1 || * || a_2 || * cos(theta)$

$< q_1, a_2 > = || a_2 || * cos(theta) = c$

Gram-Schmidt orthogonalization

Since R is upper triangular and $A = QR$ where $Q = [q_1|q_2|\ldots|q_m]$ we have

$$
\begin{aligned}
a_1 &= q_1 r_{11} \\
a_2 &= q_1 r_{12} + q_2 r_{22} \\
\vdots &= \quad \vdots \\
a_j &= q_1 r_{1j} + q_2 r_{2j} + \ldots + q_j r_{jj} \\
\vdots &= \quad \vdots \\
a_n &= q_1 r_{1n} + q_2 r_{2n} + \ldots + q_n r_{nn}
\end{aligned}
$$

From this we see that $r_{ij} = q_i^T a_j, j >= i$

Gram-Schmidt orthogonalization

```
function [Q,R] = gs_qr (A)

m = size(A,1);
n = size(A,2);

for i = 1:n
    R(i,i) = norm(A(:,i),2);
    Q(:,i) = A(:,i)./R(i,i);
    for j = i+1:n
        R(i,j) = Q(:,i)' * A(:,j);
        A(:,j) = A(:,j) - R(i,j)*Q(:,i);
    end
end

end
```

Using SVD for least squares

Recall that a singular value decomposition is given by

$$A = \begin{bmatrix} \vdots & \vdots & \vdots \\ u_1 & \dots & u_m \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \sigma_1 & & & & \\ & \ddots & & & \\ & & \sigma_r & & \\ & & & \ddots & \\ & & & & 0 \end{bmatrix} \begin{bmatrix} \dots & v_1^T & \dots \\ \dots & \vdots & \dots \\ \dots & v_n^T & \dots \end{bmatrix}$$

where σ_i are the singular values.

Using SVD for least squares

Assume that A has rank k (and hence k nonzero singular values σ_i) and recall that we want to minimize

$$||r||_2^2 = ||b - Ax||_2^2.$$

Substituting the SVD for A we find that

$$||r||_2^2 = ||b - Ax||_2^2 = ||b - USV^T x||_2^2$$

where U and V are orthogonal and S is diagonal with k nonzero singular values.

$$||b - USV^T x||_2^2 = ||U^T b - U^T USV^T x||_2^2 = ||U^T b - SV^T x||_2^2$$

Using SVD for least squares

Let $c = U^T b$ and $y = V^T x$ (and hence $x = Vy$) in $||U^T b - SV^T x||_2^2$. We now have

$$||r||_2^2 = ||c - Sy||_2^2$$

Since S has only k nonzero diagonal elements, we have

$$||r||_2^2 = \sum_{i=1}^{k} (c_i - \sigma_i y_i)^2 + \sum_{i=k+1}^{m} c_i^2$$

which is minimized when $y_i = \frac{c_i}{\sigma_i}$ for $1 \leqslant i \leqslant k$.

Using SVD for least squares

Theorem

Let A be an $m \times n$ matrix of rank r and let $A = USV^T$, the singular value decomposition. The least squares solution of the system $Ax = b$ is

$$x = \sum_{i=1}^{r} (\sigma_i^{-1} c_i) v_i$$

where $c_i = u_i^T b$.