Mathematics Methods for Computer Science

Instructor: Xubo Yang

SJTU-SE DALAB

Lecture

Designing and Analyzing Linear Systems

Theorist's Dilemma

"Find a nail for this really interesting hammer."

$$A\vec{x} = \vec{b}$$

Today's Lesson

Linear systems are insanely important.

Regression

Regression: for data analysis
Example: biological experiment
Plant growth: fertilizer, sunlight, water

Goal: predict the output of $f(\vec{x})$ for a new $\vec{x}$ without carrying out the full experiment

Linear Regression

$$f(\vec{x}) = a_1x_1 + a_2x_2 + \cdots + a_nx_n = \vec{a}^T\vec{x}$$

Find $\{a_1, \cdots, a_n\}$

n Experiments

$$\vec{x}^{(k)} \mapsto y^{(k)} \equiv f\left(\vec{x}^{(k)}\right)$$

$$\begin{gathered}
y^{(1)} = f\left(\vec{x}^{(1)}\right) = a_1 x_1^{(1)} + a_2 x_2^{(1)} + \cdots + a_n x_n^{(1)} \\
y^{(2)} = f\left(\vec{x}^{(2)}\right) = a_1 x_1^{(2)} + a_2 x_2^{(2)} + \cdots + a_n x_n^{(2)} \\
\vdots
\end{gathered}$$

Linear System for $\vec{a}$

$$\begin{pmatrix} - & \vec{x}^{(1)\top} & - \\ - & \vec{x}^{(2)\top} & - \\ & \vdots & \\ - & \vec{x}^{(n)\top} & - \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{pmatrix}$$

General Case

f can be nonlinear!

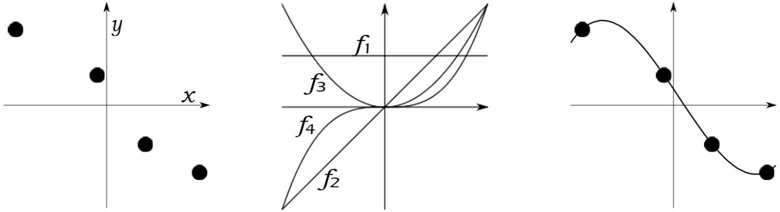
$$f(\vec{x}) = a_1 f_1(\vec{x}) + a_2 f_2(\vec{x}) + \cdots + a_m f_m(\vec{x})$$

类似于泰勒展开

$$\begin{pmatrix} f_1\left(\vec{x}^{(1)}\right) & f_2\left(\vec{x}^{(1)}\right) & \cdots & f_m\left(\vec{x}^{(1)}\right) \\ f_1\left(\vec{x}^{(2)}\right) & f_2\left(\vec{x}^{(2)}\right) & \cdots & f_m\left(\vec{x}^{(2)}\right) \\ \vdots & \vdots & \cdots & \vdots \\ f_1\left(\vec{x}^{(m)}\right) & f_2\left(\vec{x}^{(m)}\right) & \cdots & f_m\left(\vec{x}^{(m)}\right) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{pmatrix}$$

Key: write f as a linear combination of basis functions

General Case



(a) (b) (c)

Two Important Cases

(范德蒙系统)

$$f(\vec{x}) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

"Vandermonde system"

$$f(x) = acos(x + \phi)$$

Mini-Fourier

Polynomial Regression

$$f(\vec{x}) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

"Vandermonde system"

$$\begin{pmatrix} 1 & x^{(1)} & \left(x^{(1)}\right)^2 & \cdots & \left(x^{(1)}\right)^{n-1} \\ 1 & x^{(2)} & \left(x^{(2)}\right)^2 & \cdots & \left(x^{(2)}\right)^{n-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & x^{(n)} & \left(x^{(n)}\right)^2 & \cdots & \left(x^{(n)}\right)^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{pmatrix}$$

Oscillation

$$f(x) = acos(x + \phi)$$

Mini-Fourier

$$g(x) = a_1 cosx + a_2 sinx$$ (合一函数)

$$a = \sqrt{a_1^2 + a_2^2} \qquad \phi = -arctan(\frac{a2}{a1})$$

Something Fishy

Why should you have to do exactly n experiments?

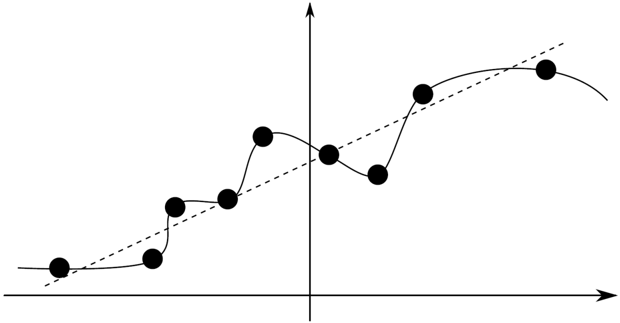
What if $y^{(k)}$ is measured with error?

Drawbacks of fitting values exactly

Overfitting noisy data

Finding patterns in statistical noise

noisy data:可以理解为测量时本身出现了误差的数据。
(对线性函数进行拟合时出现)

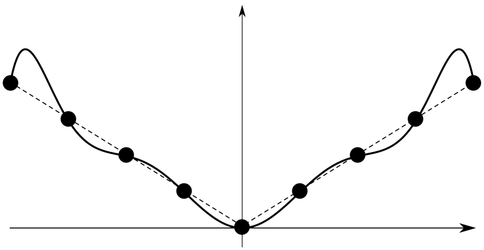


Drawbacks of fitting values exactly

Wrong basis

Basis may not be tuned to the function sampled

（对非线性函数曲线进行拟合时会出现）



Interpretation of Linear Systems

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} - & \vec{r}_1^\top & - \\ - & \vec{r}_2^\top & - \\ \vdots & \cdots & \vdots \\ - & \vec{r}_n^\top & - \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \vec{r}_1 \cdot \vec{x} \\ \vec{r}_2 \cdot \vec{x} \\ \vdots \\ \vec{r}_n \cdot \vec{x} \end{pmatrix}$$

"Guess $\vec{x}$ by observing its dot products with $\vec{r}_i$'s."

What happens when $m > n$?

(tall matrix,无解的情况最有可能发生，但是也可能唯一解或无穷多解)

Rows are likely to be incompatible.

Next best thing:

$$A\vec{x} \approx \vec{b}$$

这里使用≈而不是=
是处于对可能存在的
各种误差(数据误差、拟合函数误差)会对结果产生影响
的考虑。所以解这样的方程
就是找到误差最小值
就可以认为得到了解。

An over-determined **least-squares** problem.

Least Squares

$$A\vec{x} \approx \vec{b} \iff \min_{\vec{x}} \|A\vec{x} - \vec{b}\|_2$$

$$\iff \min_{\vec{x}} \|A\vec{x} - \vec{b}\|_2^2$$

$$\iff A^\top A\vec{x} = A^\top \vec{b}$$

(求导)

Minimizing residual square $\|A\vec{x} - \vec{b}\|_2^2$

$$\begin{aligned}
\|A\vec{x} - \vec{b}\|_2^2 &= (A\vec{x} - \vec{b}) \cdot (A\vec{x} - \vec{b}) \\
&= (A\vec{x} - \vec{b})^\top (A\vec{x} - \vec{b}) \\
&= \left(\vec{x}^\top A^\top - \vec{b}^\top\right) (A\vec{x} - \vec{b}) \\
&= \vec{x}^\top A^\top A\vec{x} - \vec{x}^\top A^\top \vec{b} - \vec{b}^\top A\vec{x} + \vec{b}^\top \vec{b} \\
&= \|A\vec{x}\|_2^2 - 2\left(A^\top \vec{b}\right) \cdot \vec{x} + \|\vec{b}\|_2^2
\end{aligned}$$

Minimum ($\nabla_{\vec{x}}$ must be zero)

$$\begin{aligned}
\vec{0} &= 2A^T A\vec{x} - 2A^T\vec{b} \\
&\implies A^\top A\vec{x} = A^\top \vec{b}
\end{aligned}$$

Normal Equations

$$A^T A\vec{x} = A^T\vec{b}$$

$A^T A$ is the Gram matrix.

Normal Equations

$$A^T A\vec{x} = A^T\vec{b}$$

$A^T A$ is the Gram matrix.

In the overdetermined case $(m > n)$, solving the least-squares problem $A\vec{x} \approx \vec{b}$ is equivalent to solving the square system $A^T A\vec{x} = A^T\vec{b}$.

Normal Equations

$$A^T A\vec{x} = A^T\vec{b}$$

$A^T A$ is the Gram matrix.

In the overdetermined case $(m > n)$, solving the least-squares problem $A\vec{x} \approx \vec{b}$ is equivalent to solving the square system $A^T A\vec{x} = A^T\vec{b}$.

How about underdetermined case $(m < n)$?

Underdetermined case

More difficult: ambiguity, too much solutions

Add additional assumptions to get a unique solution (e.g. small norm, more zeros)

Regularizing: application dependent

Methods commonly used in computer graphics, computer vision, statical analysis and machine learning

Regularization

Tikhonov regularization ("Ridge Regression;" Gaussian prior):

$$\min_{\vec{x}} \|A\vec{x} - \vec{b}\|_2^2 + \alpha\|\vec{x}\|_2^2$$

$$0 < \alpha \ll 1$$

由于待定形式的矩阵对应的解一般无穷多，所以不能直接像过定一样直接处理(因为过定条件下只要求出来的解一般不是无穷多解)，要通过加一个额外的参数来限制我们需要的解的条件。

Regularization

Tikhonov regularization
("Ridge Regression;" Gaussian prior):

$$\min_{\vec{x}} \|A\vec{x} - \vec{b}\|_2^2 + \alpha\|\vec{x}\|_2^2$$

$$0 < \alpha \ll 1$$

$$\implies \vec{0} = 2A^T A\vec{x} - 2A^T\vec{b} + 2\alpha\vec{x}$$

$$\implies (A^\top A + \alpha I_{n\times n})\vec{x} = A^\top\vec{b}$$

Example

$$\begin{pmatrix} 1 & 1 \\ 1 & 1.00001 \end{pmatrix} \vec{x} = \begin{pmatrix} 1 \\ 0.99 \end{pmatrix}$$

$$\vec{x} = (1001, -1000)$$

关于这个方程已经有一个精确解，但是为啥还要正则化的问题：虽然得到了一个确定的唯一解，但是这个唯一解与约束条件之间的数量级相差较大，在实际使用时并没有意义，(因为这一个解会使得原来的矩阵A作用退化，甚至使得方程无解)，所以我们是用正则化得到了一组近似唯一解，虽然解本身有误差，但是其实际意义更大。(当然得到的近似解的精确度与 α 的取值有关， α 越小，精确度越高)。

Example

$$\begin{pmatrix} 1 & 1 \\ 1 & 1.00001 \end{pmatrix} \vec{x} = \begin{pmatrix} 1 \\ 0.99 \end{pmatrix}$$

$$\vec{x} = (1001, -1000)$$

$$\left[\begin{pmatrix} 1 & 1 \\ 1 & 1.00001 \end{pmatrix}^\top \begin{pmatrix} 1 & 1 \\ 1 & 1.00001 \end{pmatrix} + \alpha I_{2\times 2} \right] \vec{x} = \begin{pmatrix} 1 & 1 \\ 1 & 1.00001 \end{pmatrix}^\top \begin{pmatrix} 1 \\ 0.99 \end{pmatrix}$$

$$\Longleftrightarrow \begin{pmatrix} 2+\alpha & 2.00001 \\ 2.00001 & 2.0000200001+\alpha \end{pmatrix} \vec{x} = \begin{pmatrix} 1.99 \\ 1.9900099 \end{pmatrix}$$

$$\begin{aligned} \alpha = 0.00001 &\longrightarrow \vec{x} \approx (0.499998, 0.494998) \\ \alpha = 0.001 &\longrightarrow \vec{x} \approx (0.497398, 0.497351) \\ \alpha = 0.1 &\longrightarrow \vec{x} \approx (0.485364, 0.485366) \end{aligned}$$

Regularization

Tikhonov regularization
("Ridge Regression;" Gaussian prior):

$$\min_{\vec{x}} \|A\vec{x} - \vec{b}\|_2^2 + \alpha\|\vec{x}\|_2^2$$

Lasso (Laplace prior):

$$\min_{\vec{x}} \|A\vec{x} - \vec{b}\|_2^2 + \beta\|\vec{x}\|_1$$

Elastic Net:

$$\min_{\vec{x}} \|A\vec{x} - \vec{b}\|_2^2 + \alpha\|\vec{x}\|_2^2 + \beta\|\vec{x}\|_1$$

Example: Image Alignment

A：旋转矩阵，b：平移矩阵

$$\vec{y}_k \approx A\vec{x}_k + \vec{b}, \quad A \in \mathbb{R}^{2\times 2} \quad \vec{b} \in \mathbb{R}^2$$

(a) (b) Input images with keypoints (c) Aligned images

未知数：6个(在给定的参考点数量为3时)
已知条件：应多于6个(因为是近似方程)

$$\min_{A,\vec{b}} \sum_{k=1}^{p} \left\| \left(A\vec{x}_k + \vec{b}\right) - \vec{y}_k \right\|_2^2$$

Example: Image Alignment

$$\vec{y}_k \approx A\vec{x}_k + \vec{b}$$

$\Rightarrow$Residual: $\vec{r_k} = \vec{y}_k - A\vec{x}_k - \vec{b}$ $\quad\Rightarrow\quad$ Target: $\min_{A,b} \sum_k \|\vec{r_k}\|_2^2$

$$f(A,\vec{b}) = \sum_k \left(A\vec{x}_k + \vec{b} - \vec{y}_k\right)^\top \left(A\vec{x}_k + \vec{b} - \vec{y}_k\right) \text{ since } \|\vec{v}\|_2^2 = \vec{v}^\top\vec{v}$$

$$= \sum_k \left[\vec{x}_k^\top A^\top A\vec{x}_k + 2\vec{x}_k^\top A^\top \vec{b} - 2\vec{x}_k^\top A^\top \vec{y}_k + \vec{b}^\top\vec{b} - 2\vec{b}^\top\vec{y}_k + \vec{y}_k^\top\vec{y}_k\right]$$

$$0 = \nabla_{\vec{b}} f(A,\vec{b}) = \sum_k \left[2A\vec{x}_k + 2\vec{b} - 2\vec{y}_k\right]$$

$$0 = \nabla_A f(A,\vec{b}) = \sum_k \left[2A\vec{x}_k\vec{x}_k^\top + 2\vec{b}\vec{x}_k^\top - 2\vec{y}_k\vec{x}_k^\top\right]$$

Example: Deconvolution

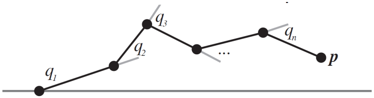
这个例子里面的
α 可以理解为
用户需要的图像清
晰的程度

(a) Sharp (b) Blurry (c) Deconvolved (d) Difference

$$\min_{\vec{x}\in\mathbb{R}^p} \|\vec{x}_0 - G\vec{x}\|_2^2$$
$$\min_{\vec{x}\in\mathbb{R}^p} \|\vec{x}_0 - G\vec{x}\|_2^2 + \alpha\|\vec{x}\|_2^2$$

Example: Robotics

Planar Serial Chain Manipulator



Problem: How to change redundant joint angles $\vec{q}$ to move toward goal position?

- Joint angles: $\vec{q} = (q_1, q_2, \cdots, q_n)^T$
- End-effector position: $\vec{p} = \begin{pmatrix} x \\ y \end{pmatrix}$
- Kinematic model: $\vec{p} = \vec{f}(\vec{q}) \xrightarrow{\text{Linearize}} \Delta\vec{p} = J\Delta\vec{q}$
- An under-determined linear least-squares problem.
- Minimum-norm solution for $\Delta\vec{q}$ given $\Delta\vec{p}$.

A Ridiculously Important Matrix

$$A^T A$$

$A^T A$ is the Gram matrix.

Properties of $A^T A$

Symmetric

B is symmetric if $B^T = B$.

Properties of $A^T A$

Symmetric

B is symmetric if $B^T = B$.

Positive (Semi-)Definite

B is positive semidefinite if for all $\vec{x} \in \mathbb{R}^n$, $\vec{x}^T B \vec{x} \geq 0$. B is positive definite if $\vec{x}^T B \vec{x} > 0$ whenever $\vec{x} \neq \vec{0}$.

Pivoting for SPD C

Goal:

Solve $C\vec{x} = \vec{d}$ for symmetric positive definite C.

$$C = \begin{pmatrix} c_{11} & \vec{v}^\top \\ \vec{v} & \tilde{C} \end{pmatrix}$$

Forward Substitution

$$E = \begin{pmatrix} 1/\sqrt{c_{11}} & \vec{0}^\top \\ \vec{r} & I_{(n-1)\times(n-1)} \end{pmatrix}$$

Symmetry Experiment

Try post-multiplication:

$$
\begin{aligned}
ECE^\top &= (EC)E^\top \\
&= \begin{pmatrix} \sqrt{c_{11}} & \vec{v}^\top/\sqrt{c_{11}} \\ \vec{0} & D \end{pmatrix} \begin{pmatrix} 1/\sqrt{c_{11}} & \vec{r}^\top \\ \vec{0} & I_{(n-1)\times(n-1)} \end{pmatrix} \\
&= \begin{pmatrix} 1 & \vec{0}^\top \\ \vec{0} & D \end{pmatrix}.
\end{aligned}
$$

What enabled this?

- Positive definite $\Rightarrow$ existance of $\sqrt{c_{11}}$
- Symmetry $\Rightarrow$ apply E to both sides

Cholesky Factorization

形式类似于LU分解，L是下三角矩阵，L^T为上三角矩阵。

$$C = LL^T$$

$$\mathsf{E}_k \cdots E_2 E_1 C E_1^\top E_2^\top \cdots E_k^\top = I_{n\times n}$$

$$\mathsf{L} = E_1^{-1} E_2^{-1} \cdots E_k^{-1}$$

Observation about Cholesky

$$L = \begin{pmatrix} L_{11} & 0 & 0 \\ \vec{\ell}_k^\top & \ell_{kk} & 0 \\ L_{31} & L_{32} & L_{33} \end{pmatrix}$$

$$\implies LL^\top = \begin{pmatrix} \times & \times & \times \\ \vec{\ell}_k^\top L_{11}^\top & \vec{\ell}_k^\top \vec{\ell}_k + \ell_{kk}^2 & \times \\ \times & \times & \times \end{pmatrix}$$

$$\ell_{kk} = \sqrt{c_{kk} - \left\|\vec{\ell}_k\right\|_2^2}$$

$$L_{11}\vec{\ell}_k = \vec{c}_k$$

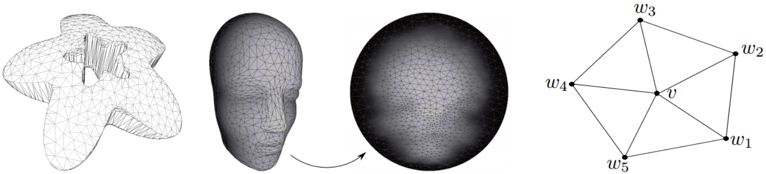
Cholesky Factorization Code

function CHOLESKY-FACTORIZATION(C)
  ▷ Factors $C = LL^T$, assuming C is symmetric and positive definite

  $L \leftarrow C$ ▷ This algorithm destructively replaces C with L
  for $k \leftarrow 1, 2, \ldots, n$
    ▷ Back-substitute to place $\vec{\ell}_k^\top$ at the beginning of row k
    for $i \leftarrow 1, \ldots, k-1$ ▷ Current element i of $\vec{\ell}_k$
      $s \leftarrow 0$
      ▷ Iterate over L_{11}; $j < i$, so the iteration maintains $L_{kj} = (\vec{\ell}_k)_j$.
      for $j \leftarrow 1, \ldots, i-1 : s \leftarrow s + L_{ij}L_{kj}$
      $L_{ki} \leftarrow {}^{(L_{ki}-s)}/_{L_{ii}}$

    ▷ Apply the formula for ℓ_{kk}
    $v \leftarrow 0$ ▷ For computing $\|\vec{\ell}_k\|_2^2$
    for $j \leftarrow 1, \ldots, k-1 : v \leftarrow v + L_{kj}^2$
    $L_{kk} \leftarrow \sqrt{L_{kk} - v}$
  return L

Harmonic Parameterization



(a) Triangle mesh (b) Parameterization (c) Harmonic condition

E.g., mesh Laplacian matrices.

Storing Sparse Matrices

(疏松矩阵)

(疏松矩阵的特点)

Want $O(n)$ storage if we have $O(n)$ nonzeros!

Examples:

- List of triplets (r,c,val)
- For each row r, matrix[r] holds a dictionary c $\rightarrow$ A[r][c]

Fill

$$\begin{pmatrix} \otimes & \times & \times & \times & \times \\ 0 & \times & 0 & 0 & 0 \\ 0 & 0 & \times & 0 & 0 \\ 0 & 0 & 0 & \times & 0 \\ 0 & 0 & 0 & 0 & \times \end{pmatrix} \Rightarrow \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix}$$

Storing Sparse Matrices

- Common strategy: Permute rows/columns
- Mostly heuristic constructions
  Minimizing fill in Cholesky is NP-complete!
- Alternative strategy:
  Avoid Gaussian elimination altogether
  Iterative solution methods - only need matrix-vector multiplication!

Banded Matrices

$$\begin{pmatrix} \times & \times & & & \\ \times & \times & \times & & \\ & \times & \times & \times & \\ & & \times & \times & \times \\ & & & \times & \times \end{pmatrix}$$

Cyclic Matrices

$$\begin{pmatrix} a & b & c & d \\ d & a & b & c \\ c & d & a & b \\ b & c & d & a \end{pmatrix}$$