5. Applications to data fitting

- dimension reduction
- rank-deficient least squares
- regularized least squares
- total least squares
- system realization

Introduction

applications in this lecture use matrices to represent *data sets:*

- a set of examples (or samples, data points, observations, measurements)
- for each example, a list of attributes or features

an $m \times n$ *data matrix* A is used to represent the data

- rows are feature vectors for m examples
- columns correspond to n features
- rows are denoted by $a_1^T, \ldots, a_m^T$ with $a_i \in \mathbf{R}^n$

Dimension reduction

low-rank approximation of data matrix can improve efficiency or performance

$$A \approx \tilde{A}Q^T \qquad \text{where } \tilde{A} \text{ is } m \times k \text{ and } Q \text{ is } n \times k$$

- we assume (without loss of generality) that Q has orthonormal columns
- columns of Q are a basis for a k-dimensional subspace in feature space $\mathbf{R}^n$
- $\tilde{A}$ is reduced data matrix; rows $\tilde{a}_i^T$ are reduced feature vectors:

$$a_i \approx Q\tilde{a}_i, \quad i = 1, \ldots, m$$

we discuss three choices for $\tilde{A}$ and Q

- truncated singular value decomposition
- truncated QR factorization
- k-means clustering

Truncated singular value decomposition

truncate SVD $A = U\Sigma V^T = \sum_i \sigma_i u_i v_i^T$ after k terms: $A \approx \tilde{A}Q^T$ with

$$\begin{aligned}\tilde{A} &= \begin{bmatrix} \sigma_1 u_1 & \sigma_2 u_2 & \cdots & \sigma_k u_k \end{bmatrix} \\ Q &= \begin{bmatrix} v_1 & v_2 & \cdots & v_k \end{bmatrix}\end{aligned}$$

- $\tilde{A}Q^T$ is the best rank-k approximation of the data matrix A (see page 4.28)

$$\tilde{A}Q^T = \sum_{i=1}^{k} \sigma_i u_i v_i^T \approx A$$

- rows $\tilde{a}_i^T$ of $\tilde{A}$ are (coordinates of) projections of the rows a_i^T on range of Q

$$\tilde{A} = \left(\sum_{i=1}^{\min\{m,n\}} \sigma_i u_i v_i^T \right) Q = AQ$$

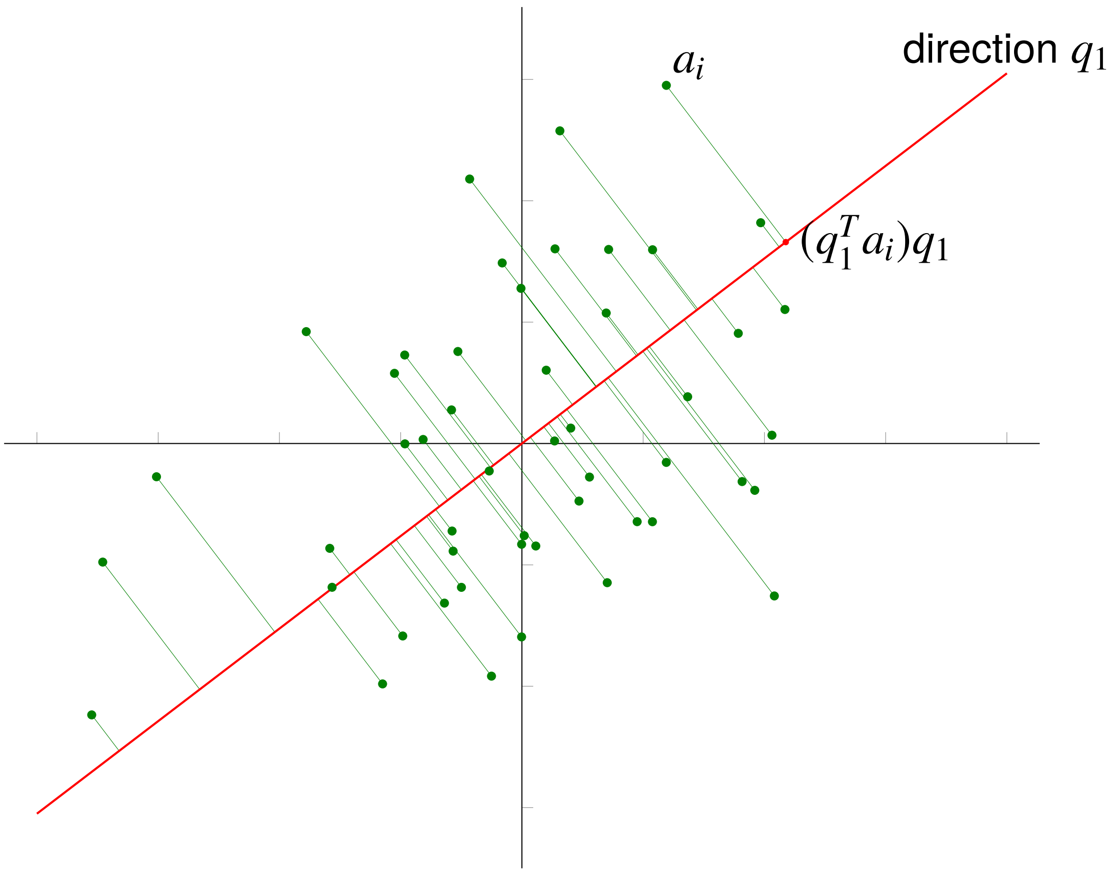
when A is centered ($\mathbf{1}^T A = 0$), columns in Q are called *principal components*

Interpretation

max–min properties of SVD give the columns of Q important optimality properties

First component: q_1 is the direction q that maximizes

$$\|Aq\|^2 = (q^T a_1)^2 + \cdots + (q^T a_m)^2$$

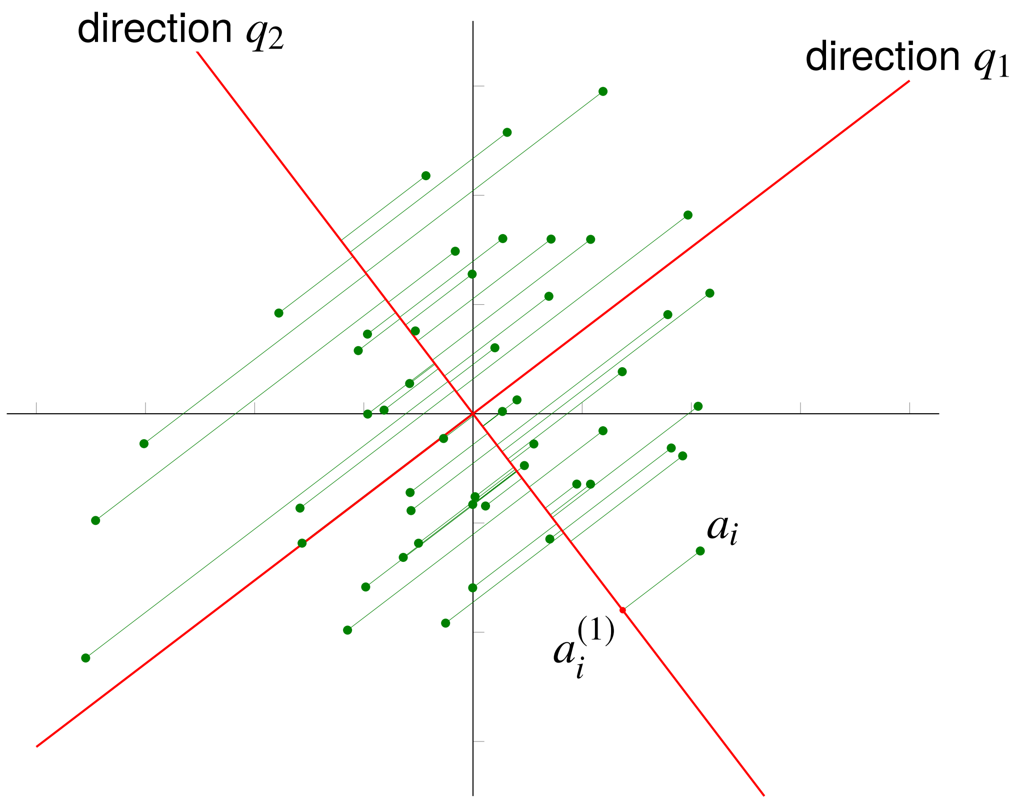


Interpretation

Second component: $q_2 = v_2$ is the first right singular vector of

$$A^{(1)} = A - \sigma_1 u_1 v_1^T = A(I - q_1 q_1^T)$$

- rows of $A^{(1)}$ are the rows of A projected on the orthogonal complement of q_1
- q_2 is the direction q that maximizes $\|A^{(1)} q\|^2$



Interpretation

Component i

$q_i = v_i$ is the first singular vector of

$$A^{(i-1)} = A - \sum_{j=1}^{i-1} \sigma_j u_j v_j^T = A(I - q_1 q_1^T - \cdots - q_{i-1} q_{i-1}^T)$$

- rows of $A^{(i-1)}$ are the rows of A projected on $\operatorname{span}\{q_1, \ldots, q_{i-1}\}^{\perp}$
- q_i is the direction q that maximizes

$$\|A^{(i-1)} q\|^2 = \left(q^T a_1^{(i-1)}\right)^2 + \left(q^T a_2^{(i-1)}\right)^2 + \cdots + \left(q^T a_m^{(i-1)}\right)^2$$

Truncated QR factorization

truncate the pivoted QR factorization of A^T after k steps

- partial QR factorization after k steps (see page 1.21)

$$PA = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} R_1^T \\ R_2^T \end{bmatrix} Q^T + \begin{bmatrix} 0 \\ B^T \end{bmatrix}, \qquad B^T Q = 0$$

  P a permutation, R_1 is $k \times k$ and upper triangular, Q has orthonormal columns

- we drop B and use the first term to define a rank-k reduced data matrix:

$$PA \approx \begin{bmatrix} R_1^T \\ R_2^T \end{bmatrix} Q^T$$

this does not have the optimality properties of the SVD but is cheaper to compute

Reduced data matrix

$$PA = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \approx \begin{bmatrix} R_1^T \\ R_2^T \end{bmatrix} Q^T$$

- $A_1 = R_1^T Q^T$: a subset of k examples from the original data matrix A
- the k-dimensional reduced feature subspace is

$$\mathrm{range}(Q) = \mathrm{range}(QR_1) = \mathrm{range}(A_1^T)$$

  reduced subspace is spanned by the feature vectors in A_1

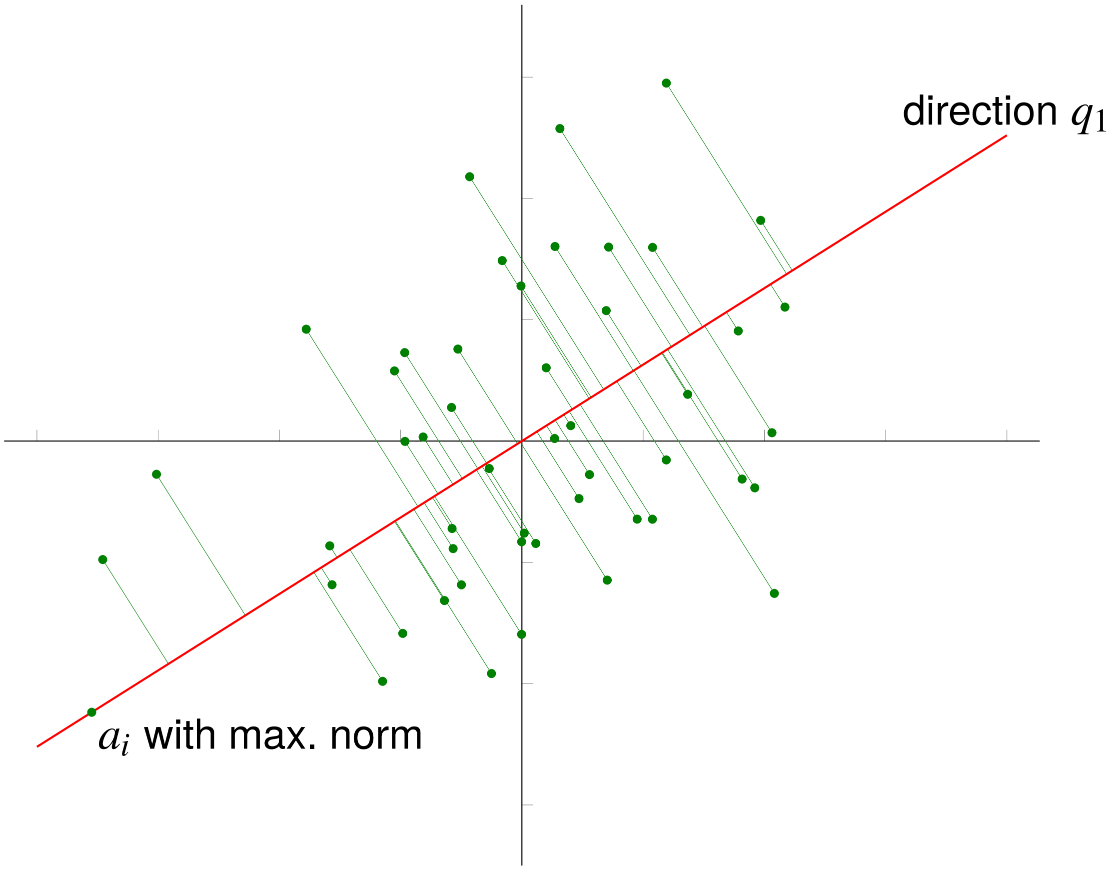
- the rows of $R_2^T Q^T$ are the rows of A_2 projected on $\mathrm{range}(Q)$:

$$A_2 Q Q^T = (R_2^T Q^T + B^T) Q Q^T = R_2^T Q^T$$

Interpretation

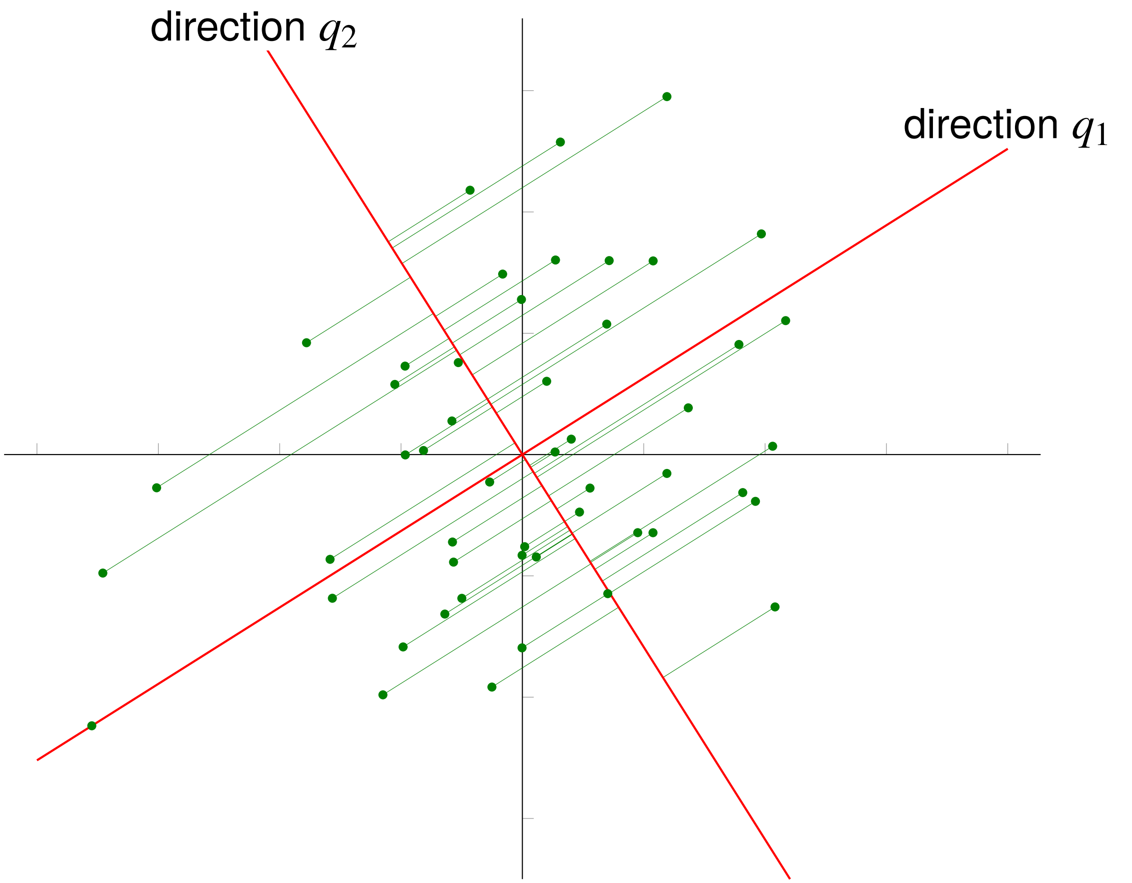
we use the pivoting rule of page 1.21

First component: q_1 is direction of largest row in A



Interpretation

Second component: q_2 is direction of largest row of $A^{(1)} = A(I - q_1 q_1^T)$



Component i: q_i is direction of largest row of

$$A^{(i-1)} = A(I - q_1 q_1^T) \cdots (I - q_{i-1} q_{i-1})^T$$

k-means clustering

run k-means on the rows of A to cluster them in k groups with representatives

$$b_1, \quad b_2, \quad \ldots, \quad b_k \in \mathbf{R}^n$$

- this can be interpreted as a rank-k approximation of A:

$$A \approx CB^T, \qquad C_{ij} = \begin{cases} 1 & \text{row } i \text{ of } A \text{ is assigned to group } j \\ 0 & \text{otherwise} \end{cases}$$

  in other words, in CB^T each row a_i^T is replaced by its group representative

- QR factorization $B = QR$ gives an orthonormal basis for $\mathrm{range}(B)$
- $\tilde{A} = CR^T$ is a possible choice of reduced data matrix
- alternatively, to improve approximation one computes $\tilde{A}$ by minimizing

$$\|A - \tilde{A}Q^T\|_F^2$$

(see homework for details)

Example: document analysis

a collection of documents is represented by a *term–document matrix* D

- each row corresponds to a word in a dictionary
- each column corresponds to a document

entries give frequencies of word in documents, usually weighted, for example, as

$$D_{ij} = f_{ij} \log(m/m_i)$$

- f_{ij} is frequency of term i in document j
- m is number of documents
- m_i is number of documents that contain term i

for consistency with the earlier notation, we define

$$A = D^T$$

A is $m \times n$ (number of documents $\times$ number of words)

Comparing documents and queries

Comparing documents: as measure of document similarity, we can use

$$\frac{a_i^T a_j}{\|a_i\| \|a_j\|}$$

- a_i^T and a_j^T are the rows of $A = D^T$ corresponding to documents i and j
- this is called the *cosine similarity:* cosine of the angle beween a_i and a_j

Query matching: find the most relevant documents based on keywords in a query

- we treat the query as a simple document, represented by an n-vector x:

$$x_j = 1 \quad \text{if term } j \text{ appears in the query,} \qquad x_j = 0 \quad \text{otherwise}$$

- we rank documents according to their cosine similiarity with x:

$$\frac{a_i^T x}{\|a_i\| \|x\|}, \quad j = 1, \ldots, m$$

Dimension reduction

it is common to make a low-rank approximation of the term–document matrix

$$D^T = A \approx \tilde{A} Q^T$$

- if the truncated SVD is used, this is called *latent semantic indexing* (LSI)
- cosine similarity of query vector x with ith row $Q\tilde{a}_i$ of reduced data matrix is

$$\frac{\tilde{a}_i^T Q^T x}{\|Q\tilde{a}_i\|\|x\|} = \frac{\tilde{a}_i^T Q^T x}{\|\tilde{a}_i\|\|x\|}$$

- an alternative is to compute the angle between $\tilde{a}_i$ and $Q^T x$:

$$\frac{\tilde{a}_i^T Q^T x}{\|\tilde{a}_i\|\|Q^T x\|}$$

References

- Lars Eldén, *Matrix Methods in Data Mining and Pattern Recognition* (2007), chapter 11.

  describes the document analysis application, including Latent Semantic Indexing and k-means clustering

- Michael W. Berry, Zlatko Drmač, Elizabeth R. Jessup, *Matrices, Vector Spaces, and Information Retrieval*, SIAM Review (1999).

  also discusses the QR factorization method

- Michael W. Berry and Murray Browne, *Understanding Search Engines: Mathematical Modeling and Text Retrieval* (2005), chapters 3 and 4.

Outline

- dimension reduction
- **rank-deficient least squares**
- regularized least squares
- total least squares
- system realization

Minimum-norm least squares solution

least squares problem with $m \times n$ matrix A and $\mathrm{rank}(A) = r$ (possibly $r < n$)

$$\text{minimize} \quad \|Ax - b\|^2$$

- on page 1.39 we showed that the minimum-norm least squares solution is

$$\hat{x} = A^\dagger b$$

- other (not minimum-norm) LS solutions are $\hat{x} + v$ for nonzero $v \in \mathrm{null}(A)$

if A has rank r and SVD $A = \sum_{i=1}^r \sigma_i u_i v_i^T$, the formulas for $A^\dagger$ and $\hat{x}$ are

$$A^\dagger = \sum_{i=1}^r \frac{1}{\sigma_i} v_i u_i^T, \qquad \hat{x} = \sum_{i=1}^r \frac{u_i^T b}{\sigma_i} v_i$$

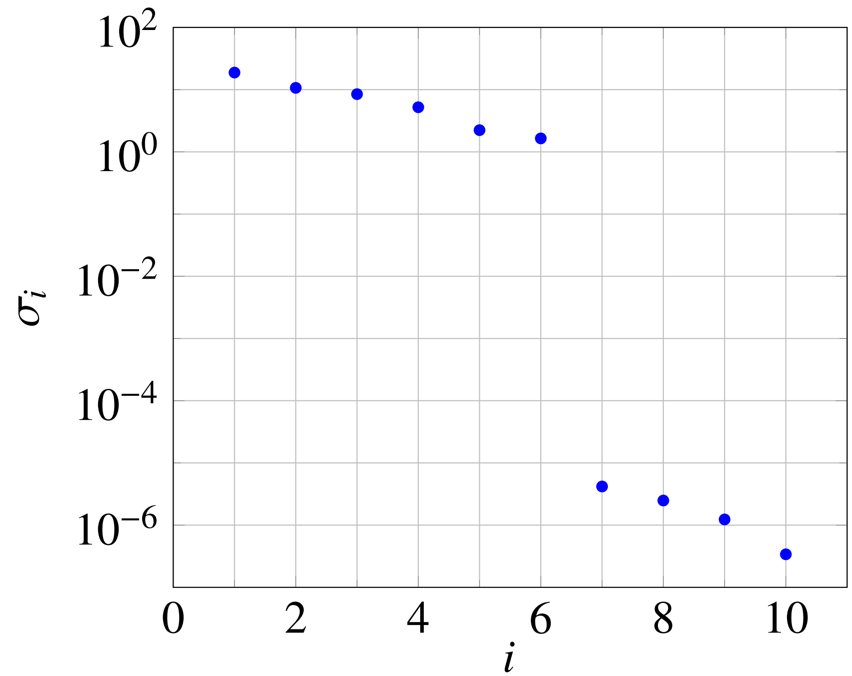
(see page 4.13 for expresson of the pseudo-inverse)

Estimating rank

a perturbation of a rank-deficient matrix will make all singular values nonzero

Example (10×10 matrix)

singular values suggest matrix is a perturbation of a matrix with rank 6



- the *numerical rank* is the number of singular values above a certain threshold
- good value of threshold is application-dependent
- truncating after numerical rank $\tilde{r}$ removes influence of small singular values

$$\hat{x} = \sum_{i=1}^{\tilde{r}} \frac{u_i^T b}{\sigma_i} v_i$$

Outline

- low-rank matrix representations
- rank-deficient least squares
- **regularized least squares**
- total least squares
- system realization

Tikhonov regularization

least squares problem with quadratic regularization

$$\text{minimize} \quad \|Ax - b\|^2 + \lambda \|x\|^2$$

- known as *Tikhonov regularization* or *ridge regression*
- weight λ controls trade-off between two objectives $\|Ax - b\|^2$ and $\|x\|^2$
- regularization term can help avoid over-fitting
- equivalent to standard least squares problem with a stacked matrix:

$$\text{minimize} \quad \left\| \begin{bmatrix} A \\ \sqrt{\lambda} I \end{bmatrix} x - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|^2$$

- for positive λ, the regularized problem always has a unique solution

$$\hat{x}_\lambda = (A^T A + \lambda I)^{-1} A^T b$$

Exercise

regularized least squares problem with a column of ones in the coefficient matrix:

$$\text{minimize} \quad \left\| \begin{bmatrix} \mathbf{1} & A \end{bmatrix} \begin{bmatrix} v \\ x \end{bmatrix} - b \right\|^2 + \lambda \|x\|^2$$

- data matrix includes a constant feature 1 (parameter v is the offset or intercept)
- associated variable v is excluded from regularization term

show that the problem is equivalent to

$$\text{minimize} \quad \|A_c x - b\|^2 + \lambda \|x\|^2$$

where A_c is the centered data matrix

$$A_c = (I - \frac{1}{m}\mathbf{1}\mathbf{1}^T)A = A - \frac{1}{m}\mathbf{1}(\mathbf{1}^T A)$$

Regularization path

suppose A has full SVD

$$A = U\Sigma V^T = \sum_{i=1}^{\min\{m,n\}} \sigma_i u_i v_i^T$$

substituting the SVD in the formula for $\hat{x}_\lambda$ shows the effect of λ:

$$\begin{aligned}
\hat{x}_\lambda = (A^T A + \lambda I)^{-1} A^T b &= (V\Sigma^T \Sigma V^T + \lambda I)^{-1} V\Sigma^T U^T b \\
&= V(\Sigma^T \Sigma + \lambda I)^{-1} V^T V \Sigma^T U^T b \\
&= V(\Sigma^T \Sigma + \lambda I)^{-1} \Sigma^T U^T b \\
&= \sum_{i=1}^{\min\{m,n\}} \frac{\sigma_i (u_i^T b)}{\sigma_i^2 + \lambda} v_i
\end{aligned}$$

this expression is valid for any matrix shape and rank

Interpretation

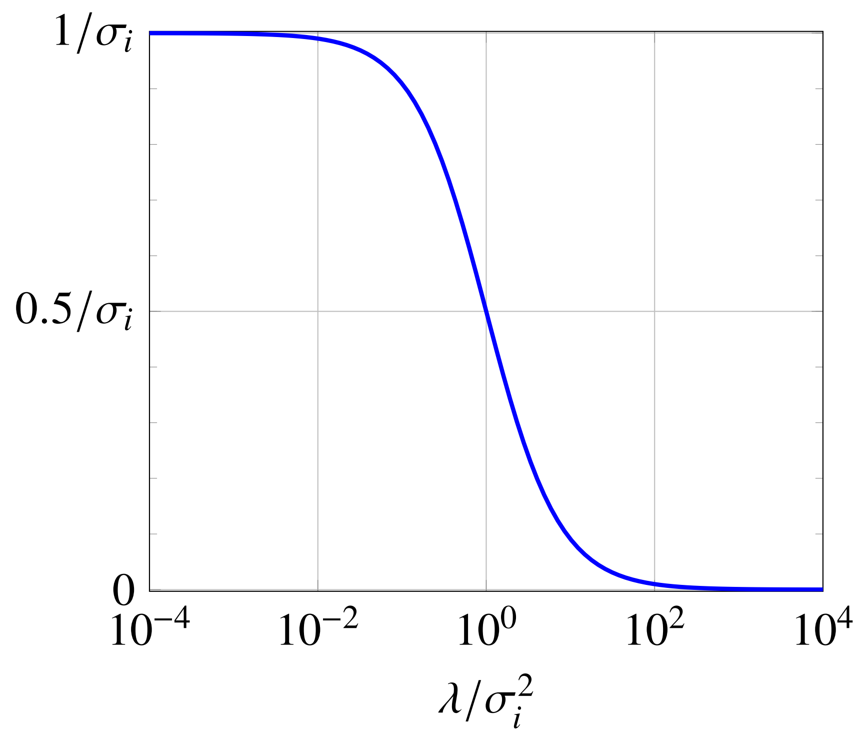
$$\hat{x}_\lambda = \sum_{i=1}^{\min\{m,n\}} \frac{\sigma_i}{\sigma_i^2 + \lambda} v_i (u_i^T b)$$

- positive λ reduces (shrinks) all terms in the sum
- terms for small σ_i are suppressed more
- all terms with $\sigma_i = 0$ are removed

plot shows the weight function

$$\frac{\sigma_i}{\sigma_i^2 + \lambda} = \frac{1/\sigma_i}{1 + \lambda/\sigma_i^2}$$

versus λ, for a term with $\sigma_i > 0$



Truncated SVD as regularization

- suppose we determine numerical rank of A by comparing σ_i with threshold τ
- truncating SVD of A gives approximation $\tilde{A} = \sum_{\sigma_i > \tau} \sigma_i u_i v_i^T$
- minimum-norm least squares solution for truncated matrix is (page 5.18)

$$\hat{x}_{\text{trunc}} = \sum_{\sigma_i > \tau} \frac{1}{\sigma_i} v_i (u_i^T b)$$

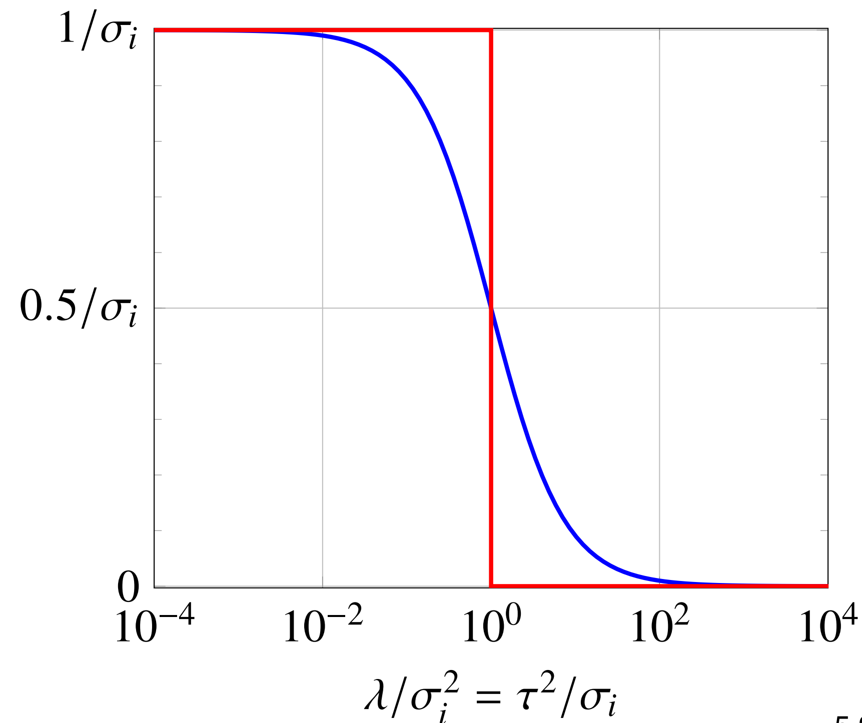
plot shows two weight functions

- Tikhonov regularization:

$$\frac{1/\sigma_i}{1 + \lambda/\sigma_i^2}$$

- truncated SVD solution with $\tau = \sqrt{\lambda}$:

$$\begin{cases} 1/\sigma_i & \sigma_i > \sqrt{\lambda} \\ 0 & \sigma_i \leq \sqrt{\lambda} \end{cases}$$



Limit for $\lambda = 0$

Regularized least squares solution

$$\hat{x}_\lambda = \sum_{i=1}^{\min\{m,n\}} \frac{\sigma_i}{\sigma_i^2 + \lambda} v_i (u_i^T b) = \sum_{i=1}^{r} \frac{\sigma_i}{\sigma_i^2 + \lambda} v_i (u_i^T b)$$

- the limit for $\lambda \to 0$ is

$$\lim_{\lambda \to 0} \hat{x}_\lambda = \sum_{i=1}^{r} \frac{1}{\sigma_i} v_i (u_i^T b)$$

- this is the minimum-norm solution of the unregularized problem (page 5.17)

Pseudo-inverse: this gives a new interpretation of the pseudo-inverse

$$\begin{aligned} A^\dagger = \sum_{i=1}^{r} \frac{1}{\sigma_i} v_i u_i^T &= \lim_{\lambda \to 0} \sum_{i=1}^{\min\{m,n\}} \frac{\sigma_i}{\sigma_i^2 + \lambda} v_i u_i^T \\ &= \lim_{\lambda \to 0} (A^T A + \lambda I)^{-1} A^T \end{aligned}$$

Example

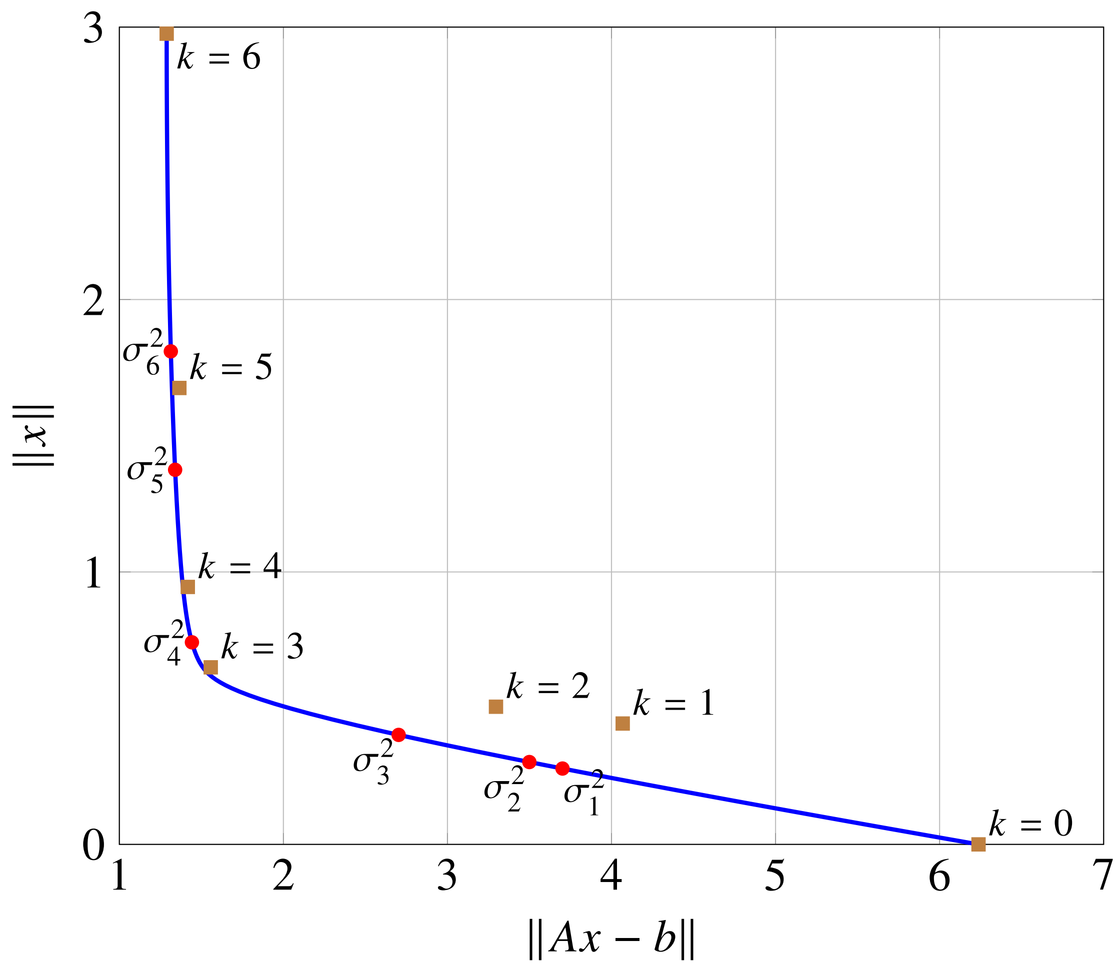
10×6 matrix with singular values

$$\sigma_1 = 10.66, \quad \sigma_2 = 9.86, \quad \sigma_3 = 7.11, \quad \sigma_4 = 0.94, \quad \sigma_5 = 0.27, \quad \sigma_6 = 0.18$$

solid line is trade-off curve

●: solution $\hat{x}_\lambda$ with $\lambda = \sigma_i^2$

■: truncate SVD after k terms



Outline

- low-rank matrix representations
- rank-deficient least squares
- regularized least squares
- **total least squares**
- system realization

Total least squares

Least squares problem

$$\text{minimize} \quad \|Ax - b\|^2$$

- can be written as constrained least squares problem with variables x and e

$$\begin{array}{ll} \text{minimize} & \|e\|^2 \\ \text{subject to} & Ax = b + e \end{array}$$

- e is the smallest adjustment to b that makes the equation $Ax = b + e$ solvable

Total least squares (TLS) problem

$$\begin{array}{ll} \text{minimize} & \|E\|_F^2 + \|e\|^2 \\ \text{subject to} & (A + E)x = b + e \end{array}$$

- variables are n-vector x, m-vector e, and $m \times n$ matrix E
- E and e are the smallest adjustments to A, b that make the equation solvable
- eliminating e gives a nonlinear LS problem: minimize $\|E\|_F^2 + \|(A + E)x - b\|^2$

TLS solution via singular value decomposition

$$\begin{array}{ll} \text{minimize} & \|E\|_F^2 + \|e\|^2 \\ \text{subject to} & (A+E)x = b+e \end{array}$$

we assume that $\sigma_{\min}(A) > \sigma_{\min}(C) > 0$ where $C = \begin{bmatrix} A & -b \end{bmatrix}$

- compute an SVD of the $m \times (n+1)$ matrix C:

$$C = \begin{bmatrix} A & -b \end{bmatrix} = \sum_{i=1}^{n+1} \sigma_i u_i v_i^T$$

- partition the right singular vector v_{n+1} of C as

$$v_{n+1} = \begin{bmatrix} w \\ z \end{bmatrix} \qquad \text{with } w \in \mathbf{R}^n \text{ and } z \in \mathbf{R}$$

- the solution of the TLS problem is

$$E = -\sigma_{n+1} u_{n+1} w^T, \qquad e = \sigma_{n+1} u_{n+1} z, \qquad x = w/z$$

Proof:

$$\begin{array}{ll} \text{minimize} & \|E\|_F^2 + \|e\|^2 \\ \text{subject to} & \begin{bmatrix} A+E & -(b+e) \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} = 0 \end{array}$$

- the matrix of rank n closest to C and its difference with C are

$$\begin{bmatrix} A+E & -(b+e) \end{bmatrix} = \sum_{i=1}^{n} \sigma_i u_i v_i^T, \qquad \begin{bmatrix} E & -e \end{bmatrix} = -\sigma_{n+1} u_{n+1} v_{n+1}^T$$

- $v_{n+1} = (w, z)$ spans the nullspace of this matrix
- if $z \neq 0$ we can normalize v_{n+1} to get a solution $x = w/z$ that satisfies

$$\begin{bmatrix} A+E & -(b+e) \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} = 0$$

- assumption $\sigma_{\min}(A) > \sigma_{\min}(C)$ implies that z is nonzero: $z = 0$ contradicts

$$\sigma_{\min}(A) = \min_{\|y\|=1} \|Ay\| > \sigma_{\min}(C) = \|Aw - bz\|$$

Extension

$$\begin{array}{ll} \text{minimize} & \|E\|_F^2 + \|e\|^2 \\ \text{subject to} & A_1 x_1 + (A_2 + E)x_2 = b + e \end{array} \qquad (1)$$

- variables are E, e, x_1, x_2
- we make the smallest adjustment to A_2 and b that makes the equation solvable
- no adjustment is made to A_1
- eliminating e gives a nonlinear least squares problem in E, x_1, x_2:

$$\text{minimize} \quad \|E\|_F^2 + \|A_1 x_1 + (A_2 + E)x_2 - b\|^2$$

- we will assume that A_1 has linearly independent columns

Solution

- assume A_1 has QR factorization $A_1 = Q_1 R$ and $Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ is orthogonal
- multiply the constraint in (1) on the left with Q^T:

$$Rx_1 + (Q_1^T A_2 + E_1)x_2 = Q_1^T b + e_1, \qquad (Q_2^T A_2 + E_2)x_2 = Q_2^T b + e_2 \tag{2}$$

  where $E_1 = Q_1^T E, \;\; E_2 = Q_2^T E, \;\; e_1 = Q_1^T e, \;\; e_2 = Q_2^T e$

- cost function in (1) is

$$\|E\|_F^2 + \|e\|^2 = \|E_1\|_F^2 + \|E_2\|_F^2 + \|e_1\|^2 + \|e_2\|^2$$

- first equation in (2) is always solvable, so $E_1 = 0$, $e_1 = 0$ are optimal
- for the 2nd equation we solve the TLS problem in E_2, e_2, x_2:

$$\begin{array}{ll} \text{minimize} & \|E_2\|_F^2 + \|e_2\|^2 \\ \text{subject to} & (Q_2^T A_2 + E_2)x_2 = Q_2^T b + e_2 \end{array}$$

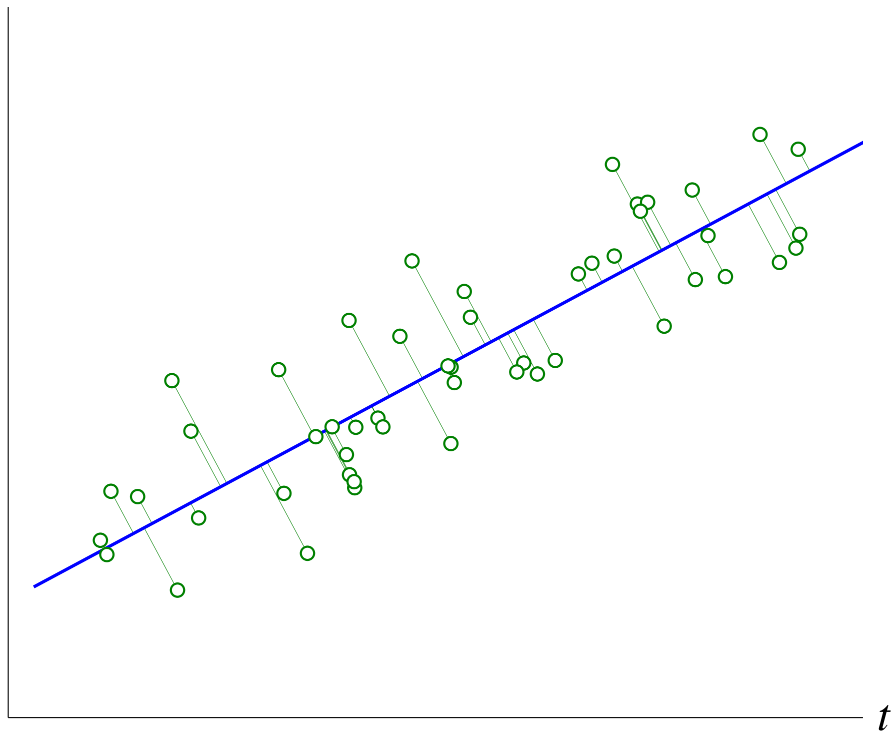
- we compute x_1 from x_2 by solving $Rx_1 = Q_1^T b - Q_1^T A_2 x_2$

Example: orthogonal distance regression

fit an affine function $f(t) = x_1 + x_2 t$ to m points (a_i, b_i)

$$\begin{array}{ll} \text{minimize} & \|\delta a\|^2 + \|\delta b\|^2 \\ \text{subject to} & x_1 \mathbf{1} + x_2(a + \delta a) = b + \delta b \end{array}$$

- the variables are m-vectors δa, δb and scalars x_1, x_2
- we fit the line by minimizing the sum of squared distances to the line



Outline

- low-rank matrix representations
- rank-deficient least squares
- regularized least squares
- total least squares
- **system realization**

Linear dynamical system

State space model

$$
\begin{aligned}
x(t+1) &= Ax(t) + Bu(t) \\
y(t) &= Cx(t) + Du(t)
\end{aligned}
$$

$u(t) \in \mathbf{R}^m$ is the input, $y(t) \in \mathbf{R}^p$ is the output, $x(t) \in \mathbf{R}^n$ is the state at time t

Input–output model

- $y(t)$ is a linear function of the past inputs

$$
\begin{aligned}
y(t) &= Du(t) + CBu(t-1) + CABu(t-2) + CA^2Bu(t-3) + \cdots \\
&= H_0u(t) + H_1u(t-1) + H_2u(t-2) + H_3u(t-3) + \cdots
\end{aligned}
$$

  where we define $H_0 = D$ and $H_k = CA^{k-1}B$ for $k \geq 1$

- the matrices H_k are the *impulse response coefficients* or *Markov parameters*

From past inputs to future outputs

suppose the inputs $u(t)$ is zero for $t > 0$ and $x(-M) = 0$

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(T) \end{bmatrix} = \begin{bmatrix} H_0 & H_1 & H_2 & \cdots & H(-M) \\ H_1 & H_2 & H_3 & \cdots & H(-M+1) \\ H_2 & H_3 & H_4 & \cdots & H(-M+2) \\ \vdots & \vdots & \vdots & & \vdots \\ H_T & H_{T+1} & H_{T+2} & \cdots & H(T-M) \end{bmatrix} \begin{bmatrix} u(0) \\ u(-1) \\ u(-2) \\ \vdots \\ u(-M) \end{bmatrix}$$

- matrix of impulse response coefficients maps past inputs to future outputs
- coefficient matrix is a block-Hankel matrix (constant on antidiagonals)

System realization problem

find state space model A, B, C, D from observed $H_0, H_1, \ldots, H_N$

- if the impulse response coefficients $H_1, \ldots, H_N$ are exact,

$$\begin{bmatrix} H_1 & H_2 & \cdots & H_{N-k+1} \\ H_2 & H_3 & \cdots & H_{N-k+2} \\ \vdots & \vdots & & \vdots \\ H_k & H_{k+1} & \cdots & H_N \end{bmatrix} = \begin{bmatrix} CB & CAB & \cdots & CA^{N-k}B \\ CAB & CA^2B & \cdots & CA^{N-k+1}B \\ \vdots & \vdots & \cdots & \vdots \\ CA^{k-1}B & CA^kB & \cdots & CA^{N-1}B \end{bmatrix}$$

$$= \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{k-1} \end{bmatrix} \begin{bmatrix} B & AB & \cdots & A^{N-k}B \end{bmatrix}$$

- block Hankel matrix of impulse response coefficients has rank n
- from a rank-n factorization, we can compute A, B, C (and D from $D = H_0$)

System realization with inexact data

- estimate system order from singular values of block Hankel matrix
- truncate SVD to find approximate rank-n factorization

$$\begin{bmatrix} H_1 & H_2 & \cdots & H_{N-k+1} \\ H_2 & H_3 & \cdots & H_{N-k+2} \\ \vdots & \vdots & & \vdots \\ H_k & H_{k+1} & \cdots & H_N \end{bmatrix} \approx \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_k \end{bmatrix} \begin{bmatrix} V_1 & V_2 & \cdots & V_{N-k+1} \end{bmatrix}$$

- find A, B, C that approximately satisfy $U_i = CA^{i-1}$ and $V_j = A^{j-1}B$
- for example, take $C = U_1$, $B = V_1$, and A from the least squares problem

$$\text{minimize} \quad \left\| \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_{k-1} \end{bmatrix} A - \begin{bmatrix} U_2 \\ U_3 \\ \vdots \\ U_k \end{bmatrix} \right\|_F^2$$