13. Cholesky factorization

- positive definite matrices
- examples
- Cholesky factorization
- complex positive definite matrices
- kernel methods
- pivoted Cholesky factorization
Definitions

- a symmetric matrix \( A \in \mathbb{R}^{n \times n} \) is positive semidefinite if
  \[
  x^T Ax \geq 0 \quad \text{for all } x
  \]

- a symmetric matrix \( A \in \mathbb{R}^{n \times n} \) is positive definite if
  \[
  x^T Ax > 0 \quad \text{for all } x \neq 0
  \]

this is a subset of the positive semidefinite matrices

note: if \( A \) is symmetric and \( n \times n \), then \( x^T Ax \) is the function

\[
 x^T Ax = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij}x_i x_j = \sum_{i=1}^{n} A_{ii}x_i^2 + 2 \sum_{i>j} A_{ij}x_i x_j
\]

this is called a \textit{quadratic form}
Example

\[
A = \begin{bmatrix}
9 & 6 \\
6 & a
\end{bmatrix}
\]

\[
x^T Ax = 9x_1^2 + 12x_1x_2 + ax_2^2 = (3x_1 + 2x_2)^2 + (a - 4)x_2^2
\]

- \( A \) is positive definite for \( a > 4 \)
  \[
x^T Ax > 0 \quad \text{for all nonzero } x
\]

- \( A \) is positive semidefinite but not positive definite for \( a = 4 \)
  \[
x^T Ax \geq 0 \quad \text{for all } x, \quad x^T Ax = 0 \quad \text{for } x = (2, -3)
\]

- \( A \) is not positive semidefinite for \( a < 4 \)
  \[
x^T Ax < 0 \quad \text{for } x = (2, -3)
\]
Simple properties

- every positive definite matrix $A$ is nonsingular
  
  $$Ax = 0 \implies x^T Ax = 0 \implies x = 0$$
  
  (last step follows from positive definiteness)

- every positive definite matrix $A$ has positive diagonal elements
  
  $$A_{ii} = e_i^T A e_i > 0$$

- every positive semidefinite matrix $A$ has nonnegative diagonal elements
  
  $$A_{ii} = e_i^T A e_i \geq 0$$
Schur complement

partition \( n \times n \) symmetric matrix \( A \) as

\[
A = \begin{bmatrix}
    A_{11} & A_{2:n,1}^T \\
    A_{2:n,1} & A_{2:n,2:n}
\end{bmatrix}
\]

- the Schur complement of \( A_{11} \) is defined as the \( (n - 1) \times (n - 1) \) matrix

\[
S = A_{2:n,2:n} - \frac{1}{A_{11}} A_{2:n,1} A_{2:n,1}^T
\]

- if \( A \) is positive definite, then \( S \) is positive definite

  to see this, take any \( x \neq 0 \) and define \( y = -(A_{2:n,1}^T x)/A_{11} \); then

\[
x^T S x = \begin{bmatrix} y \\ x \end{bmatrix}^T \begin{bmatrix}
    A_{11} & A_{2:n,1}^T \\
    A_{2:n,1} & A_{2:n,2:n}
\end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} > 0
\]

  because \( A \) is positive definite
Singular positive semidefinite matrices

- we have seen that positive definite matrices are nonsingular (page 13.4)
- if $A$ is positive semidefinite, but not positive definite, then it is singular

To see this, suppose $A$ is positive semidefinite but not positive definite

- there exists a nonzero $x$ with $x^T Ax = 0$
- since $A$ is positive semidefinite the following function is nonnegative:

\[
f(t) = (x - tAx)^T A(x - tAx)
\]

\[
= x^T Ax - 2tx^T A^2 x + t^2 x^T A^3 x
\]

\[
= -2t \|Ax\|^2 + t^2 x^T A^3 x
\]

- $f(t) \geq 0$ for all $t$ is only possible if $\|Ax\| = 0$; therefore $Ax = 0$
- hence there exists a nonzero $x$ with $Ax = 0$, so $A$ is singular
Exercises

• show that if $A \in \mathbb{R}^{n \times n}$ is positive semidefinite, then

$$B^T AB$$

is positive semidefinite for any $B \in \mathbb{R}^{n \times m}$

• show that if $A \in \mathbb{R}^{n \times n}$ is positive definite, then

$$B^T AB$$

is positive definite for any $B \in \mathbb{R}^{n \times m}$ with linearly independent columns
Outline

• positive definite matrices
• examples
• Cholesky factorization
• complex positive definite matrices
• kernel methods
• pivoted Cholesky factorization
Exercise: resistor circuit

\[ y_1 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} R_1 + R_3 & R_3 \\ R_3 & R_2 + R_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \]

show that the matrix

\[ A = \begin{bmatrix} R_1 + R_3 & R_3 \\ R_3 & R_2 + R_3 \end{bmatrix} \]

is positive definite if \( R_1, R_2, R_3 \) are positive.
Solution

Solution from physics

- $x^T A x = y^T x$ is the power delivered by sources, dissipated by resistors
- power dissipated by the resistors is positive unless both currents are zero

Algebraic solution

\[
\begin{align*}
x^T A x &= (R_1 + R_3)x_1^2 + 2R_3x_1x_2 + (R_2 + R_3)x_2^2 \\
&= R_1x_1^2 + R_2x_2^2 + R_3(x_1 + x_2)^2 \\
&\geq 0
\end{align*}
\]

and $x^T A x = 0$ only if $x_1 = x_2 = 0$
Gram matrix

recall the definition of Gram matrix of a matrix $B$ (page 4.20):

$$A = B^T B$$

- every Gram matrix is positive semidefinite

$$x^T A x = x^T B^T B x = \|B x\|^2 \geq 0 \quad \forall x$$

- a Gram matrix is positive definite if

$$x^T A x = x^T B^T B x = \|B x\|^2 > 0 \quad \forall x \neq 0$$

in other words, $B$ has linearly independent columns
Graph Laplacian

recall definition of node-arc incidence matrix of a directed graph (page 3.29)

\[
B_{ij} = \begin{cases} 
  1 & \text{if edge } j \text{ ends at vertex } i \\
  -1 & \text{if edge } j \text{ starts at vertex } i \\
  0 & \text{otherwise}
\end{cases}
\]

assume there are no self-loops and at most one edge between any two vertices

\[
B = \begin{bmatrix}
  -1 & -1 & 0 & 1 & 0 \\
   1 & 0 & -1 & 0 & 0 \\
   0 & 0 & 1 & -1 & -1 \\
   0 & 1 & 0 & 0 & 1 \\
\end{bmatrix}
\]
Graph Laplacian

the positive semidefinite matrix $A = BB^T$ is called the *Laplacian* of the graph

$$A_{ij} = \begin{cases} 
\text{degree of vertex } i & \text{if } i = j \\
-1 & \text{if } i \neq j \text{ and vertices } i \text{ and } j \text{ are adjacent} \\
0 & \text{otherwise}
\end{cases}$$

the degree of a vertex is the number of edges incident to it

$$A = BB^T = \begin{bmatrix}
3 & -1 & -1 & -1 \\
-1 & 2 & -1 & 0 \\
-1 & -1 & 3 & -1 \\
-1 & 0 & -1 & 2
\end{bmatrix}$$
Laplacian quadratic form

recall the interpretation of matrix–vector multiplication with $B^T$ (page 3.31)

- if $y$ is vector of node potentials, then $B^T y$ contains potential differences:

\[
(B^T y)_j = y_k - y_l \quad \text{if edge } j \text{ goes from vertex } l \text{ to } k
\]

- $y^T Ay = y^T B B^T y$ is the sum of squared potential differences

\[
y^T Ay = \|B^T y\|^2 = \sum_{\text{edges } i \rightarrow j} (y_j - y_i)^2
\]

this is also known as the Dirichlet energy function

**Example:** for the graph on the previous page

\[
y^T A y = (y_2 - y_1)^2 + (y_4 - y_1)^2 + (y_3 - y_2)^2 + (y_1 - y_3)^2 + (y_4 - y_3)^2
\]
Weighted graph Laplacian

- we associate a nonnegative weight $w_k$ with edge $k$
- the weighted graph Laplacian is the matrix $L = A \text{diag}(w)A^T$

$$L_{ij} = \begin{cases} \sum_{k \in \mathcal{N}_i} w_k & \text{if } i = j \quad \text{(where } \mathcal{N}_i \text{ are the edges incident to vertex } i) \\ -w_k & \text{if } i \neq j \text{ and edge } k \text{ is between vertices } i \text{ and } j \\ 0 & \text{otherwise} \end{cases}$$

$$L = \begin{bmatrix} w_1 + w_2 + w_4 & -w_1 & -w_4 & -w_2 \\ -w_1 & w_1 + w_3 & -w_3 & 0 \\ -w_4 & -w_3 & w_3 + w_4 + w_5 & -w_5 \\ -w_2 & 0 & -w_5 & w_2 + w_5 \end{bmatrix}$$

this is the conductance matrix of a resistive circuit ($w_k$ is conductance in branch $k$)
Outline

- positive definite matrices
- examples
- Cholesky factorization
- complex positive definite matrices
- kernel methods
- pivoted Cholesky factorization
Cholesky factorization

every positive definite matrix $A \in \mathbb{R}^{n \times n}$ can be factored as

$$A = R^T R$$

where $R$ is upper triangular with positive diagonal elements

- complexity of computing $R$ is $(1/3)n^3$ flops
- $R$ is called the Cholesky factor of $A$
- can be interpreted as “square root” of a positive definite matrix
- gives a practical method for testing positive definiteness
Cholesky factorization algorithm

\[
\begin{bmatrix}
A_{11} & A_{1,2:n} \\
A_{2:n,1} & A_{2:n,2:n}
\end{bmatrix}
= \begin{bmatrix}
R_{11} & 0 \\
R^T_{1,2:n} & R^T_{2:n,2:n}
\end{bmatrix}\begin{bmatrix}
R_{11} & R_{1,2:n} \\
0 & R_{2:n,2:n}
\end{bmatrix}
= \begin{bmatrix}
R_{11}^2 & R_{11}R_{1,2:n} \\
R_{11}R^T_{1,2:n} & R^T_{1,2:n}R_{1,2:n} + R^T_{2:n,2:n}R_{2:n,2:n}
\end{bmatrix}
\]

1. compute first row of \( R \):

\[R_{11} = \sqrt{A_{11}}, \quad R_{1,2:n} = \frac{1}{R_{11}}A_{1,2:n}\]

2. compute 2, 2 block \( R_{2:n,2:n} \) from

\[A_{2:n,2:n} - R^T_{1,2:n}R_{1,2:n} = R^T_{2:n,2:n}R_{2:n,2:n}\]

this is a Cholesky factorization of order \( n - 1 \)
Discussion

the algorithm works for positive definite $A$ of size $n \times n$

- step 1: if $A$ is positive definite then $A_{11} > 0$
- step 2: if $A$ is positive definite, then

$$A_{2:n,2:n} - R_{1,2:n}^T R_{1,2:n} = A_{2:n,2:n} - \frac{1}{A_{11}} A_{2:n,1} A_{2:n,1}^T$$

is positive definite (see page 13.5)

- hence the algorithm works for $n = m$ if it works for $n = m - 1$
- it obviously works for $n = 1$; therefore it works for all $n$
Example

\[
\begin{bmatrix}
25 & 15 & -5 \\
15 & 18 & 0 \\
-5 & 0 & 11 \\
\end{bmatrix}
= \begin{bmatrix}
R_{11} & 0 & 0 \\
R_{12} & R_{22} & 0 \\
R_{13} & R_{23} & R_{33} \\
\end{bmatrix}
\begin{bmatrix}
R_{11} & R_{12} & R_{13} \\
0 & R_{22} & R_{23} \\
0 & 0 & R_{33} \\
\end{bmatrix}
\]

\[
= \begin{bmatrix}
5 & 0 & 0 \\
3 & 3 & 0 \\
-1 & 1 & 3 \\
\end{bmatrix}
\begin{bmatrix}
5 & 3 & -1 \\
0 & 3 & 1 \\
0 & 0 & 3 \\
\end{bmatrix}
\]
Example

\[
\begin{bmatrix}
25 & 15 & -5 \\
15 & 18 & 0 \\
-5 & 0 & 11
\end{bmatrix}
= \begin{bmatrix} R_{11} & 0 & 0 \\
R_{12} & R_{22} & 0 \\
R_{13} & R_{23} & R_{33} \end{bmatrix}
\begin{bmatrix} R_{11} & R_{12} & R_{13} \\
R_{12} & R_{22} & R_{23} \\
0 & 0 & R_{33} \end{bmatrix}
\]

- first row of \( R \)

\[
\begin{bmatrix}
25 & 15 & -5 \\
15 & 18 & 0 \\
-5 & 0 & 11
\end{bmatrix}
= \begin{bmatrix} 5 & 0 & 0 \\
3 & R_{22} & 0 \\
-1 & R_{23} & R_{33} \end{bmatrix}
\begin{bmatrix} 5 & 3 & -1 \\
0 & R_{22} & R_{23} \\
0 & 0 & R_{33} \end{bmatrix}
\]

- second row of \( R \)

\[
\begin{bmatrix}
18 & 0 \\
0 & 11
\end{bmatrix}
- \begin{bmatrix} 3 \\
-1
\end{bmatrix}
\begin{bmatrix} 3 & -1
\end{bmatrix}
= \begin{bmatrix} R_{22} & 0 \\
R_{23} & R_{33} \end{bmatrix}
\begin{bmatrix} R_{22} & R_{23} \\
0 & R_{33} \end{bmatrix}
\]

\[
\begin{bmatrix} 9 & 3 \\
3 & 10 \end{bmatrix}
= \begin{bmatrix} 3 & 0 \\
1 & R_{33} \end{bmatrix}
\begin{bmatrix} 3 & 1 \\
0 & R_{33} \end{bmatrix}
\]

- third column of \( R \): \( 10 - 1 = R_{33}^2 \), i.e., \( R_{33} = 3 \)
Solving equations with positive definite $A$

solve $Ax = b$ with $A$ a positive definite $n \times n$ matrix

**Algorithm**

- factor $A$ as $A = R^T R$
- solve $R^T R x = b$
  - solve $R^T y = b$ by forward substitution
  - solve $Rx = y$ by back substitution

**Complexity:** $(1/3)n^3 + 2n^2 \approx (1/3)n^3$ flops

- factorization: $(1/3)n^3$
- forward and backward substitution: $2n^2$
Cholesky factorization of Gram matrix

- suppose $B$ is an $m \times n$ matrix with linearly independent columns
- the Gram matrix $A = B^T B$ is positive definite (page 4.20)

two methods for computing the Cholesky factor of $A$, given $B$

1. compute $A = B^T B$, then Cholesky factorization of $A$

   $$ A = R^T R $$

2. compute QR factorization $B = QR$; since

   $$ A = B^T B = R^T Q^T Q R = R^T R $$

   the matrix $R$ is the Cholesky factor of $A$
Example

\[
B = \begin{bmatrix}
3 & -6 \\
4 & -8 \\
0 & 1 \\
\end{bmatrix}, \quad A = B^T B = \begin{bmatrix}
25 & -50 \\
-50 & 101 \\
\end{bmatrix}
\]

1. Cholesky factorization:

\[
A = \begin{bmatrix}
5 & 0 \\
-10 & 1 \\
\end{bmatrix} \begin{bmatrix}
5 & -10 \\
0 & 1 \\
\end{bmatrix}
\]

2. QR factorization

\[
B = \begin{bmatrix}
3 & -6 \\
4 & -8 \\
0 & 1 \\
\end{bmatrix} = \begin{bmatrix}
3/5 & 0 \\
4/5 & 0 \\
0 & 1 \\
\end{bmatrix} \begin{bmatrix}
5 & -10 \\
0 & 1 \\
\end{bmatrix}
\]
Comparison of the two methods

**Numerical stability:** QR factorization method is more stable

- see the example on page 8.16
- QR method computes $R$ without “squaring” $B$ (i.e., forming $B^T B$)
- this is important when the columns of $B$ are “almost” linearly dependent

**Complexity**

- method 1: cost of symmetric product $B^T B$ plus Cholesky factorization

  $$mn^2 + (1/3)n^3 \text{ flops}$$

- method 2: $2mn^2$ flops for QR factorization

- method 1 is faster but only by a factor of at most two (if $m \gg n$)
Sparse positive definite matrices

Cholesky factorization of dense matrices

- $(1/3)n^3$ flops
- on a standard computer: a few seconds or less, for $n$ up to several 1000

Cholesky factorization of sparse matrices

- if $A$ is very sparse, $R$ is often (but not always) sparse
- if $R$ is sparse, the cost of the factorization is much less than $(1/3)n^3$
- exact cost depends on $n$, number of nonzero elements, sparsity pattern
- very large sets of equations can be solved by exploiting sparsity
Sparse Cholesky factorization

if $A$ is sparse and positive definite, it is usually factored as

$$A = PR^TRP^T$$

$P$ a permutation matrix; $R$ upper triangular with positive diagonal elements

**Interpretation**: we permute the rows and columns of $A$ and factor

$$P^TAP = R^TR$$

- choice of permutation greatly affects the sparsity $R$
- there exist several heuristic methods for choosing a good permutation
Example

sparsity pattern of $A$

pattern of $P^TAP$

Cholesky factor of $A$

Cholesky factor of $P^TAP$
Solving sparse positive definite equations

solve $Ax = b$ with $A$ a sparse positive definite matrix

Algorithm

1. compute sparse Cholesky factorization $A = PR^T R^T P^T$
2. permute right-hand side: $c := P^T b$
3. solve $R^T y = c$ by forward substitution
4. solve $Rz = y$ by back substitution
5. permute solution: $x := Pz$
Outline

- positive definite matrices
- examples
- Cholesky factorization
- complex positive definite matrices
- kernel methods
- pivoted Cholesky factorization
Quadratic form

suppose $A$ is $n \times n$ and Hermitian ($A_{ij} = \bar{A}_{ji}$)

$$x^H A x = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} \bar{x}_i x_j$$

$$= \sum_{i=1}^{n} A_{ii} |x_i|^2 + \sum_{i>j} (A_{ij} \bar{x}_i x_j + \bar{A}_{ij} x_i \bar{x}_j)$$

$$= \sum_{i=1}^{n} A_{ii} |x_i|^2 + 2 \text{Re} \sum_{i>j} A_{ij} \bar{x}_i x_j$$

note that $x^H A x$ is real for all $x \in \mathbb{C}^n$
Complex positive definite matrices

- A Hermitian $n \times n$ matrix $A$ is positive semidefinite if

\[ x^H A x \geq 0 \quad \text{for all } x \in \mathbb{C}^n \]

- A Hermitian $n \times n$ matrix $A$ is positive definite if

\[ x^H A x > 0 \quad \text{for all nonzero } x \in \mathbb{C}^n \]

Cholesky factorization

Every positive definite matrix $A \in \mathbb{C}^{n\times n}$ can be factored as

\[ A = R^H R \]

Where $R$ is upper triangular with positive real diagonal elements.
Outline

- positive definite matrices
- examples
- Cholesky factorization
- complex positive definite matrices
- kernel methods
- pivoted Cholesky factorization
Regularized least squares model fitting

- we revisit the data fitting problem with linear-in-parameters model (page 9.9)

\[
\hat{f}(x) = \theta_1 f_1(x) + \theta_2 f_2(x) + \cdots + \theta_p f_p(x)
\]

\[
= \theta^T F(x)
\]

- \( F(x) = (f_1(x), \ldots, f_p(x)) \) is a \( p \)-vector of basis functions \( f_1(x), \ldots, f_p(x) \)

Regularized least squares model fitting (page 10.7)

\[
\text{minimize} \quad \sum_{k=1}^N \left( \theta^T F(x^{(k)}) - y^{(k)} \right)^2 + \lambda \sum_{j=1}^p \theta_j^2
\]

- \( (x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)}) \) are \( N \) examples

- to simplify notation, we add regularization for all coefficients \( \theta_1, \ldots, \theta_p \)

- next discussion can be modified to handle \( f_1(x) = 1 \), regularization \( \sum_{j=2}^p \theta_j^2 \)
Regularized least squares problem in matrix notation

\[ \text{minimize} \quad \| A \theta - b \|^2 + \lambda \| \theta \|^2 \]

- \( A \) has size \( N \times p \) (number of examples \( \times \) number of basis functions)

\[
A = \begin{bmatrix}
F(x^{(1)})^T \\
F(x^{(2)})^T \\
\vdots \\
F(x^{(N)})^T 
\end{bmatrix} = \begin{bmatrix}
f_1(x^{(1)}) & f_2(x^{(1)}) & \cdots & f_p(x^{(1)}) \\
f_1(x^{(2)}) & f_2(x^{(2)}) & \cdots & f_p(x^{(2)}) \\
\vdots & \vdots & \ddots & \vdots \\
f_1(x^{(N)}) & f_2(x^{(N)}) & \cdots & f_p(x^{(N)})
\end{bmatrix}
\]

- \( b \) is the \( N \)-vector \( b = (y^{(1)}, \ldots, y^{(N)}) \)

- we discuss methods for problems with \( N \ll p \) (\( A \) is very wide)

- the equivalent “stacked” least squares problem (p.10.3) has size \((p + N) \times p\)

- QR factorization method may be too expensive when \( N \ll p \)
Solution of regularized LS problem

from the normal equations:

\[ \hat{\theta} = (A^T A + \lambda I)^{-1} A^T b = A^T (A A^T + \lambda I)^{-1} b \]

- second expression follows from the “push-through” identity

\[ (A^T A + \lambda I)^{-1} A^T = A^T (A A^T + \lambda I)^{-1} \]

this is easily proved, by writing it as \( A^T (A A^T + \lambda I) = (A^T A + \lambda I) A^T \)

- from the second expression for \( \hat{\theta} \) and the definition of \( A \),

\[ \hat{f}(x) = \hat{\theta}^T F(x) = w^T A F(x) = \sum_{i=1}^{N} w_i F(x^{(i)})^T F(x) \]

where \( w = (A A^T + \lambda I)^{-1} b \)
Algorithm

1. compute the \( N \times N \) matrix \( Q = AA^T \), which has elements

\[
Q_{ij} = F(x^{(i)})^T F(x^{(j)}), \quad i, j = 1, \ldots, N
\]

2. use a Cholesky factorization to solve the equation

\[
(Q + \lambda I)w = b
\]

Remarks

• \( \hat{\theta} = A^T w \) is not needed; \( w \) is sufficient to evaluate the function \( \hat{f}(x) \):

\[
\hat{f}(x) = \sum_{i=1}^{N} w_i F(x^{(i)})^T F(x)
\]

• complexity: \( (1/3)N^3 \) flops for factorization plus cost of computing \( Q \)
Example: multivariate polynomials

\( \hat{f}(x) \) is a polynomial of degree \( d \) (or less) in \( n \) variables \( x = (x_1, \ldots, x_n) \)

- \( \hat{f}(x) \) is a linear combination of all possible monomials

\[
    x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n}
\]

where \( k_1, \ldots, k_n \) are nonnegative integers with \( k_1 + k_2 + \cdots + k_n \leq d \)

- number of different monomials is

\[
    \binom{n + d}{n} = \frac{(n + d)!}{n! \ d!}
\]

Example: for \( n = 2, \ d = 3 \) there are ten monomials

\[
    1, \ x_1, \ x_2, \ x_1^2, \ x_1x_2, \ x_2^2, \ x_1^3, \ x_1^2x_2, \ x_1x_2^2, \ x_2^3
\]
Multinomial formula

\[(x_0 + x_1 + \cdots + x_n)^d = \sum_{k_0 + \cdots + k_n = d} \frac{(d + 1)!}{k_0! k_1! \cdots k_n!} x_0^{k_0} x_1^{k_1} \cdots x_n^{k_n}\]

sum is over all nonnegative integers \(k_0, k_1, \ldots, k_n\) with sum \(d\)

- setting \(x_0 = 1\) gives

\[ (1 + x_1 + x_2 + \cdots + x_n)^d = \sum_{k_1 + \cdots + k_n \leq d} c_{k_1k_2\cdots k_n} x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n}\]

- the sum includes all monomials of degree \(d\) or less with variables \(x_1, \ldots, x_n\)

- coefficient \(c_{k_1k_2\cdots k_n}\) is defined as

\[ c_{k_1k_2\cdots k_n} = \frac{(d + 1)!}{k_0! k_1! k_2! \cdots k_n!} \quad \text{with} \quad k_0 = d - k_1 - \cdots - k_n \]
Vector of monomials

write polynomial of degree $d$ or less, with variables $x \in \mathbb{R}^n$, as

$$\hat{f}(x) = \theta^T F(x)$$

- $F(x)$ is vector of basis functions

$$\sqrt{c_{k_1\ldots k_n}} x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n} \quad \text{for all } k_1 + k_2 + \cdots + k_n \leq d$$

- length of $F(x)$ is $p = (n + d)!/(n! \cdot d!)$

- multinomial formula gives simple formula for inner products $F(u)^T F(v)$:

$$F(u)^T F(v) = \sum_{k_1 + \cdots + k_n \leq d} c_{k_1\ldots k_n} (u_1^{k_1} \cdots u_n^{k_n})(v_1^{k_1} \cdots v_n^{k_n})$$

$$= \left(1 + u_1 v_1 + \cdots + u_n v_n\right)^d$$

- only $2n + 1$ flops needed for inner product of length $p = (n + d)!/(n! \cdot d!)$
Example

vector of monomials of degree $d = 3$ or less in $n = 2$ variables

\[
F(u)^T F(v) = \begin{bmatrix}
1 \\
\sqrt{3}u_1 \\
\sqrt{3}u_2 \\
\sqrt{3}u_1^2 \\
\sqrt{3}u_2^2 \\
\sqrt{6}u_1u_2 \\
\sqrt{3}u_1^2 \\
u_1^3 \\
\sqrt{3}u_2^2 \\
\sqrt{3}u_1u_2 \\
\sqrt{3}u_2^2 \\
u_2^3 \\
u_1^3 \\
u_2^3
\end{bmatrix}^T \begin{bmatrix}
1 \\
\sqrt{3}v_1 \\
\sqrt{3}v_2 \\
\sqrt{3}v_1^2 \\
\sqrt{3}v_2^2 \\
\sqrt{6}v_1v_2 \\
\sqrt{3}v_1^2 \\
v_1^3 \\
\sqrt{3}v_2^2 \\
\sqrt{3}v_1v_2 \\
\sqrt{3}v_2^2 \\
v_2^3 \\
v_1^3 \\
v_2^3
\end{bmatrix}
\]

\[= (1 + u_1v_1 + u_2v_2)^3\]
Least squares fitting of multivariate polynomials

fit polynomial of \( n \) variables, degree \( \leq d \), to points \( (x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)}) \)

**Algorithm** (see page 13.33)

1. compute the \( N \times N \) matrix \( Q \) with elements

\[
Q_{ij} = K(x^{(i)}, x^{(j)}) \quad \text{where} \quad K(u, v) = (1 + u^T v)^d
\]

2. use a Cholesky factorization to solve the equation \((Q + \lambda I)w = b\)

- the fitted polynomial is

\[
\hat{f}(x) = \sum_{i=1}^{N} w_i K(x^{(i)}, x) = \sum_{i=1}^{N} w_i (1 + (x^{(i)})^T x)^d
\]

- complexity: \( nN^2 \) flops for computing \( Q \), plus \( (1/3)N^3 \) for the factorization, \( i.e. \),

\[
nN^2 + (1/3)N^3 \text{ flops}
\]
Kernel methods

Kernel function: a generalized inner product $K(u, v)$

- $K(u, v)$ is inner product of vectors of basis functions $F(u)$ and $F(v)$
- $F(u)$ may be infinite-dimensional
- kernel methods work with $K(u, v)$ directly, do not require $F(u)$

Examples

- the polynomial kernel function $K(u, v) = (1 + u^T v)^d$
- the Gaussian radial basis function kernel

$$K(u, v) = \exp \left( -\frac{||u - v||^2}{2\sigma^2} \right)$$

- kernels exist for computing with graphs, texts, strings of symbols, ...
Example: handwritten digit classification

we apply the method of page 13.38 to least squares classification

- training set is 10000 images from MNIST data set ($\approx 1000$ examples per digit)
- vector $x$ is vector of pixel intensities (size $n = 28^2 = 784$)
- we use the polynomial kernel with degree $d = 3$:

$$K(u, v) = (1 + u^T v)^3$$

hence $F(z)$ has length $p = (n + d)!/(n!d!) = 80931145$

- we calculate ten Boolean classifiers

$$\hat{f}_k(x) = \text{sign}(\tilde{f}_k(x)), \quad k = 1, \ldots 10$$

$\hat{f}_k(x)$ distinguishes digit $k - 1$ (outcome $+1$) form other digits (outcome $-1$)

- the Boolean classifiers are combined in the multi-class classifier

$$\hat{f}(x) = \arg\max_{k=1,\ldots,10} \tilde{f}_k(x)$$
Least squares Boolean classifier

**Algorithm:** compute Boolean classifier for digit $k - 1$ versus the rest

1. compute $N \times N$ matrix $Q$ with elements

$$Q_{ij} = (1 + (x^{(i)})^T x^{(j)})^d, \quad i, j = 1, \ldots, N$$

2. define $N$-vector $b = (y^{(1)}, \ldots, y^{(N)})$ with elements

$$y^{(i)} = \begin{cases} +1 & x^{(i)} \text{ is an example of digit } k - 1 \\ -1 & \text{otherwise} \end{cases}$$

3. solve the equation $(Q + \lambda I)w = b$

the solution $w$ gives the Boolean classifier for digit $k - 1$ versus rest

$$\tilde{f}_k(x) = \sum_{i=1}^{N} w_i (1 + (x^{(i)})^T x)^d$$
Complexity

- the matrix $Q$ is the same for each of the ten Boolean classifiers
- hence, only the right-hand side of the equation

\[(Q + \lambda I)w = y^d\]

is different for each Boolean classifier

Complexity

- constructing $Q$ requires $N^2/2$ inner products of length $n$: $nN^2$ flops
- Cholesky factorization of $Q + \lambda I$: $(1/3)N^3$ flops
- solve the equation $(Q + \lambda I)w = y^d$ for the 10 right-hand sides: $20N^2$ flops
- total is $(1/3)N^3 + nN^2$
percentage of misclassified digits versus $\lambda$
### Confusion matrix

<table>
<thead>
<tr>
<th>Digit</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>965</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>8</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>980</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1127</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1135</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>2</td>
<td>988</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>16</td>
<td>8</td>
<td>1</td>
<td>1032</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>973</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>8</td>
<td>6</td>
<td>4</td>
<td>1010</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>957</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>14</td>
<td>982</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>874</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>892</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>4</td>
<td>0</td>
<td>5</td>
<td>2</td>
<td>937</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>958</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>13</td>
<td>13</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>987</td>
<td>2</td>
<td>7</td>
<td>1028</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>11</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>934</td>
<td>6</td>
<td>974</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>7</td>
<td>13</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>966</td>
<td>1009</td>
</tr>
<tr>
<td>All</td>
<td>990</td>
<td>1155</td>
<td>1015</td>
<td>1002</td>
<td>986</td>
<td>897</td>
<td>964</td>
<td>1028</td>
<td>964</td>
<td>999</td>
<td>10000</td>
</tr>
</tbody>
</table>

- Multiclass classifier ($\lambda = 10^4$) on 10000 test examples
- 292 digits are misclassified (2.9% error)
Examples of misclassified digits

<table>
<thead>
<tr>
<th>Digit</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>!0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>!1</td>
<td>!1</td>
<td>!1</td>
<td></td>
<td></td>
<td></td>
<td>!1</td>
<td>!1</td>
<td>!1</td>
</tr>
<tr>
<td>2</td>
<td>!2</td>
<td>!2</td>
<td></td>
<td>!2</td>
<td>!2</td>
<td>!2</td>
<td>!2</td>
<td>!2</td>
<td>!2</td>
<td>!2</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Cholesky factorization
Examples of misclassified digits

<table>
<thead>
<tr>
<th>Digit</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
</table>
Outline

- positive definite matrices
- examples
- Cholesky factorization
- complex positive definite matrices
- kernel methods
- pivoted Cholesky factorization
Pivoted Cholesky factorization

the following factorization exists for positive semidefinite $A$

\[ A = P^T R^T R P \]

- $P$ is a permutation matrix
- $R$ is $r \times n$, leading $r \times r$ submatrix is upper triangular with positive diagonal:

\[
R = \begin{bmatrix}
R_{11} & R_{12} & \cdots & R_{1r} & R_{1,r+1} & \cdots & R_{1n} \\
0 & R_{22} & \cdots & R_{2r} & R_{2,r+1} & \cdots & R_{2n} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & R_{rr} & R_{r,r+1} & \cdots & R_{rn}
\end{bmatrix}
\]

- can be chosen to satisfy $R_{11} \geq R_{22} \geq \cdots \geq R_{rr} > 0$
- $A = (RP)^T (RP)$ is a full-rank factorization of $A$ (page 4.34)
- $r$ is the rank of $A$ ($r = n$ for positive definite $A$)
- if $A$ is positive definite, reduces to standard Cholesky factorization of $PAP^T$
(Standard) Colesky factorization

in the algorithm on p.13.16 for Cholesky factorization $A = R^T R$:

- after $k$ steps we have completed a partial factorization

$$A = \begin{bmatrix}
    R_{11} & \cdots & 0 \\
    \vdots & \ddots & \vdots \\
    R_{1k} & \cdots & R_{kk} \\
    \hline
    R_{1,k+1} & \cdots & R_{k,k+1} \\
    \vdots & \ddots & \vdots \\
    R_{1n} & \cdots & R_{kn}
\end{bmatrix}
\begin{bmatrix}
    I \\
    0
\end{bmatrix}
\begin{bmatrix}
    I & 0 \\
    0 & S_k
\end{bmatrix}
\begin{bmatrix}
    R_{11} & \cdots & R_{1k} & R_{1,k+1} & \cdots & R_{1n} \\
    \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
    0 & \cdots & R_{kk} & R_{k,k+1} & \cdots & R_{kn}
\end{bmatrix}
\begin{bmatrix}
    I \\
    0
\end{bmatrix}$$

- row $k + 1$ of $R$ and the matrix $S_{k+1}$ are found from the equality

$$S_k = \begin{bmatrix}
    R_{k+1,k+1} & 0 \\
    \hline
    R_{k+1,(k+2):n}^T & I
\end{bmatrix}
\begin{bmatrix}
    1 & 0 \\
    0 & S_{k+1}
\end{bmatrix}
\begin{bmatrix}
    R_{k+1,k+1} & R_{k+1,(k+2):n} \\
    \hline
    0 & I
\end{bmatrix}$$
Pivoted Cholesky factorization algorithm

the algorithm is readily extended to compute the pivoted Cholesky factorization

\[ A = P^T R^T RP \]

- after \( k \) steps we have computed a partial factorization

\[
P_k A P_k^T = \begin{bmatrix}
R_{1:k,1:k}^T & 0 \\
R_{1:k,(k+1):n}^T & I
\end{bmatrix}
\begin{bmatrix}
I & 0 \\
0 & S_k
\end{bmatrix}
\begin{bmatrix}
R_{1:k,1:k} & R_{1:k,(k+1):n} \\
0 & I
\end{bmatrix}
\]

- we start the algorithm with \( P_0 = I \) and \( S_0 = A \)
- if \( S_k = 0 \), the algorithm terminates with \( r = k \)
- before step \( k + 1 \) we reorder \( S_k \) to move largest diagonal element to position 1,1
- this reordering requires modifying \( P_k \) and reordering the columns of \( R_{1:k,(k+1):n} \)
**Example**

\[
A = \begin{bmatrix}
A_{11} & A_{12} & A_{13} & A_{14} \\
A_{21} & A_{22} & A_{23} & A_{24} \\
A_{31} & A_{32} & A_{33} & A_{34} \\
A_{41} & A_{42} & A_{43} & A_{44}
\end{bmatrix} = \begin{bmatrix}
6 & 3 & 10 & -1 \\
3 & 18 & 15 & 0 \\
10 & 15 & 25 & -5 \\
-1 & 0 & -5 & 6
\end{bmatrix}
\]

**Step 1**

- apply symmetric reordering to move \( A_{33} \) to the 1,1 position
- find first row of \( R \) and \( S_1 \)

\[
\begin{bmatrix}
A_{33} & A_{31} & A_{32} & A_{34} \\
A_{13} & A_{11} & A_{12} & A_{14} \\
A_{23} & A_{21} & A_{22} & A_{24} \\
A_{43} & A_{41} & A_{42} & A_{44}
\end{bmatrix} = \begin{bmatrix}
25 & 10 & 15 & -5 \\
10 & 6 & 3 & -1 \\
15 & 3 & 18 & 0 \\
-5 & -1 & 0 & 6
\end{bmatrix}
\]

\[
\begin{bmatrix}
5 & 0 & 0 & 0 \\
2 & 1 & 0 & 0 \\
3 & 0 & 1 & 0 \\
-1 & 0 & 0 & 1
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 2 & -3 & 1 \\
0 & -3 & 9 & 3 \\
0 & 1 & 3 & 5
\end{bmatrix} = \begin{bmatrix}
5 & 2 & 3 & -1 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
Example

Step 2

- move second diagonal element of $S_1$ to first position
- compute second row of $R$ and $S_2$

\[
\begin{bmatrix}
A_{33} & A_{32} & A_{31} & A_{34} \\
A_{23} & A_{22} & A_{21} & A_{24} \\
A_{13} & A_{12} & A_{11} & A_{14} \\
A_{43} & A_{42} & A_{41} & A_{44}
\end{bmatrix}
\]

\[
\begin{bmatrix}
5 & 0 & 0 & 0 \\
3 & 1 & 0 & 0 \\
2 & 0 & 1 & 0 \\
-1 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 9 & -3 & 3 \\
0 & -3 & 2 & 1 \\
0 & 3 & 1 & 5
\end{bmatrix}
\begin{bmatrix}
5 & 3 & 2 & -1 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
\begin{bmatrix}
5 & 0 & 0 & 0 \\
3 & 3 & 0 & 0 \\
2 & -1 & 1 & 0 \\
-1 & 1 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 2 \\
0 & 0 & 2 & 4
\end{bmatrix}
\begin{bmatrix}
5 & 3 & 2 & -1 \\
0 & 3 & -1 & 1 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

Cholesky factorization
**Example**

**Step 3**
- move second diagonal element of $S_2$ to first position
- compute third row of $R$ and $S_3$

\[
\begin{bmatrix}
A_{33} & A_{32} & A_{34} & A_{31} \\
A_{23} & A_{22} & A_{24} & A_{21} \\
A_{43} & A_{42} & A_{44} & A_{41} \\
A_{13} & A_{12} & A_{14} & A_{11}
\end{bmatrix}
\]

\[
= \begin{bmatrix}
5 & 0 & 0 & 0 \\
3 & 3 & 0 & 0 \\
-1 & 1 & 1 & 0 \\
2 & -1 & 0 & 1
\end{bmatrix} \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 4 & 2 \\
0 & 0 & 2 & 1
\end{bmatrix} \begin{bmatrix}
5 & 3 & -1 & 2 \\
0 & 3 & 1 & -1 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
= \begin{bmatrix}
5 & 0 & 0 & 0 \\
3 & 3 & 0 & 0 \\
-1 & 1 & 2 & 0 \\
2 & -1 & 1 & 1
\end{bmatrix} \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix} \begin{bmatrix}
5 & 3 & -1 & 2 \\
0 & 3 & 1 & -1 \\
0 & 0 & 2 & 1 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
**Example**

**Result:** since $S_3$ is zero, the algorithm terminates with $r = 3$ and the factorization

\[
\begin{bmatrix}
A_{33} & A_{32} & A_{34} & A_{31} \\
A_{23} & A_{22} & A_{24} & A_{21} \\
A_{43} & A_{42} & A_{44} & A_{41} \\
A_{13} & A_{12} & A_{14} & A_{11}
\end{bmatrix}
= \begin{bmatrix}
25 & 15 & -5 & 10 \\
15 & 18 & 0 & 3 \\
-5 & 0 & 6 & -1 \\
10 & 3 & -1 & 6
\end{bmatrix}
= \begin{bmatrix}
5 & 0 & 0 \\
3 & 3 & 0 \\
-1 & 1 & 2 \\
2 & -1 & 1
\end{bmatrix}
\begin{bmatrix}
5 & 3 & -1 & 2 \\
0 & 3 & 1 & -1 \\
0 & 0 & 2 & 1
\end{bmatrix}

Cholesky factorization 13.53
Factorization theorem for positive semidefinite matrices

A positive semidefinite $n \times n$ matrix $A$ has rank $r$ if and only if it can be factored as

$$A = BB^T$$

where $B$ is $n \times r$ with linearly independent columns.

- the pivoted Cholesky factorization proves the “only if” part
- “if” statement follows from page 4.33
- other factorization algorithms exist (symmetric eigendecomposition)