1 Linear Algebra

Vector Properties

Linear independence — Linear combination $Au = u_1 a_1 + ... + u_n a_n = \sum_{i=1}^n u_i a_i$ only has unique solutions for u (*unique representation theorem*), if $Au = 0$ then $u = 0$, and A is full rank

Unit vector — $u = \frac{\tilde{u}}{||\tilde{u}||}$, therefore $||u||^2 = 1$

Inner product — $u \cdot v = u^\top v = \sum_{i=1}^n u_i v_i = cos(\varphi) \, ||u|| \, ||v||$ — Properties:

- $u \cdot v = v \cdot u$
- $(u+v) \cdot w = u \cdot w + v \cdot w$
- $(\alpha u) \cdot v = \alpha(u \cdot v)$
- Positive definite: $u \cdot u \geq 0$
- $u \cdot u = 0 \Leftrightarrow u = 0_v$

Norm — $||u|| = \sqrt{u \cdot u}$ — Properties:

- $||u+v|| = ||u|| + ||v||$
- $||\alpha u|| = |\alpha| ||u||$

Cauchy Schwarz inequality — $||u \cdot v|| \leq ||u|| \, ||v||$ with equality iff $\varphi = 0$ i.e. $u = \alpha v$ or if u or $v = 0_v$

Proof:

- First direction of proof: If $u = \alpha v$ or u or $v = 0_v$, we can show that the equality holds
- Second direction of proof: If $u \neq \alpha v$ or u and $v \neq 0_v$, we can show that the inequality cannot hold:
  - u can be decomposed into $u_v + u_{v\perp}$
  - Then, we have $||u \cdot v|| = ||(u_v + u_{v\perp}) \cdot v|| = ||u_v|| \cdot ||v||$
  - Based on Pythagorean theorem, we know that $||u||^2 > ||u_v||^2$
  - Then, we have $||u \cdot v|| = ||u_v|| \cdot ||v|| < ||u|| \cdot ||v||$

Triangle inequality — $||u+v|| \leq ||u|| + ||v||$ with equality iff $\varphi = 0$ i.e. $u = \alpha v$ or if u or $v = 0_v$

Orthogonal vectors — Properties:

- $u \cdot v = 0$
- $||u+v||^2 = ||u||^2 + ||v||^2$
- *Pythagorean theorem*: $||u-v||^2 = ||u||^2 + ||v||^2$
- Non-zero pairwise orthogonal vectors u_n and u_m are linearly independent
  Proof:
  - Let $\sum_n \alpha_n u_n = 0_v$
  - Then, $= 0_v \cdot u_m = (\sum_n \alpha_n u_n) \cdot u_m = \sum_n \alpha_n (u_n) \cdot u_m) = \alpha_m ||u_m||^2$
  - Then, $\alpha_m = 0$ for all m, meaning that all u_m are linearly independent

Orthonormal vectors — Vectors are orthonormal iff $||u|| = ||v|| = 1$ and $u \cdot v = 0$

Projection — Projection of $v \in V$ onto $s \in S$ given by: $v_S = \frac{v \cdot s}{||s||^2} s = (v \cdot s)s$ if s is a unit vector

Vector Spaces

Vector space V — Properties:

- Additive closure: If $u, v \in V$ then $u + v \in V$
- Scalar closure: If $u \in V$ then $\alpha u \in V$
- $\exists 0_v$ such that $u + 0_v = u$
- $\exists -u$ such that $u + -u = 0_v$
- $u + v = v + u$
- $(u+v)+w = u+(v+w)$
- $\alpha(\beta u) = (\alpha\beta)u$
- $\alpha(u+v) = \alpha u + \alpha v$
- $u(\alpha+\beta) = \alpha u + \beta u$

Subspace S — Properties: S is a subspace of V iff:

- $0_v \in S$
- Additive closure
- Scalar closure

Proof:

- If S is a subspace of V subspace properties immediately follow
- If subspace properties are satisfied for S, S must be a subspace of V because operations are inherited (for addition, multiplication) resp. can be derived from subspace properties (for $0_V, -v$)

Invariant subspace H — H is an invariant subspace of S spanned by S if $Sh \in H$ for all $h \in H$ — Properties:

- S has an eigenvector in H
- If S is symmetric, $H^\perp$ is also an invariant subspace of S

Orthogonal complement $S^\perp$ — Subspace, composed of set of vectors that are orthogonal to S — Properties:

- The intersection of S and $S^\perp$ is $\{0_v\}$
- $\dim(S) + \dim(S^\perp) = \dim(V)$

Span — Span of $\{s_i\}_{i=1}^n$ is the set of all vectors that can be expressed as a linear combination of $\{s_i\}_{i=1}^n$. $\sum_{i=1}^n u_i s_i$
Span of matrix A is the span of its column vectors. $Au = u_1 a_1 + ... + u_n a_n = \sum_{i=1}^n u_i a_i$
A span is a subspace, since for a linear combination, we can derive additive closure and scalar closure.

(Orthonormal) basis — Unique set of all (orthonormal) vectors that are linearly independent and span the whole of a subspace.

- *Orthonormal representation theorem*: Any vector $x \in S$ can be expressed in terms of orthonormal basis: $x = \sum_i (x \cdot s_i) s_i$
- *Parveval's theorem*: Extension of orthonormal representation theorem: $x \cdot y = \sum_i (x \cdot s_i)(y \cdot s_i)$
- *Gram Schmidt orthonormalization*: Procedure to generate orthonormal basis $\{s_i\}_{i=1}^n$ from linearly independent vectors $\{x^{(i)}\}_{i=1}^n$:
  - $\tilde{s_1} = x_1$
  - $\tilde{s_k} = x_k - \sum_{i=1}^{k-1} (x_k \cdot s_1) s_1$ for $k > 1$
  - $s_i = \frac{\tilde{s_i}}{||\tilde{s_i}||}$

Dimension — Number of vectors in basis of S.

Orthogonal vectors in spaces —

- Let S be spanned by orthonormal $s_1, s_2, ...$ and $v \in V$
- *Orthogonal decomposition theorem*: $v = v_S + v_{S\perp}$ where $v \in V$, $v_S \in S$ and $v_{S\perp} \in S^\perp$
- *Orthogonality principle*
  - v_S is the projection of $v \in V$ to S iff $(v - v_S) \cdot s_i = 0$
  - This can be rewritten to linear equation system $v \cdot s_i = v_S \cdot s_i = \sum_k \alpha_k (s_k \cdot s_i)$ since $v_S = \sum_k \alpha_k s_k$
- $v_{S\perp} = v - v_S = v - \sum_k (v \cdot s_k) s_k$
- *Approximation in a subspace theorem*:
  - Unique best representation of v in S is given by projection of v to S: $||v - s'|| \geq ||v - v_S||$ for some arbitrary $s' \in S$
  - Any subset U of S is closest to v iff it is closest to v_S
    Proof:
    - $\text{argmin}_u ||v - u|| = \text{argmin}_u ||v - u||^2 = \text{argmin}_u ||v_S + v_{S\perp} - u||^2 = \text{argmin}_u ||v_S - u||^2 + ||v_{S\perp}||^2$ given Pythagorean theorem $= \text{argmin}_u ||v_S - u||^2$

Linear Equations

Let $Xb = y$ where $X \in \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^m$, $y \in \mathbb{R}^n$ and b is unknown

- Number of distinct equations = Number of linearly independent rows in $[X|b] = \text{rank}([X|b]) \leq \min(n, m+1)$
- Number of LHS solutions should = Number of RHS solutions = $\text{rank}(X) \leq \min(n, m)$

Solutions:

- If $\text{rank}(X) < \text{rank}([X|b])$, system is inconsistent (no solution)
- If $\text{rank}(X) = \text{rank}([X|b]) < m$, system is singular (infinitely many solutions) and underdetermined because we have fewer distinct equations than unknowns
- If $\text{rank}(X) = \text{rank}([X|b]) = m = n$, system is non-singular (exactly one solution) and exactly determined
- If $\text{rank}(X) = \text{rank}([X|b]) = m < n$, system is non-singular and overdetermined

General Matrix Properties

Matrices —

- $A \in \mathbb{R}^{n \times m}$ with elements A_{ij}, rows $i = 1, ..., n$, columns $j = 1, ..., m$
- Transpose $A^\top$
- Identity matrix I with 1 on diagonal, 0 elsewhere
- Scalar matrix K with k on diagonal, 0 elsewhere

Operations —

- Element-wise addition: Returns matrix of same size
- Element-wise scalar multiplication: Returns matrix of same size
- Matrix multiplication:
  - $A^{n \times p} B^{p \times m} = C^{n \times m}$
    - $r_v \times c_v = s$
    - $c_v \times r_v = M$
    - $M \times c_v = c_v$
    - $r_v \times M = r_v$
    - $M \times M = M$
  - Element in C is sum-product of row in A and column in B: $C_{ij} = A^{(i)} \cdot B^{(j)}$
  - Column vector in C is a linear combination of the columns in A: $C^{(j)} = AB^{(j)} = \sum_p A^{(j=p)} b_p^{(j)}$
  - Row vector in C is a linear combination of the rows in B: $C^{(i)} = A^{(i)} B = \sum_p a_p^{(i)} B^{(i=p)}$
  - $C = A[B^{(j=1)}|...|B^{(j=m)}]$
  - $C = [A^{(i=1)}|...|A^{(i=n)}]^\top B = [A^{(i=1)}B|...|A^{(i=n)}B]^\top$

Impliations —

- $Ae_k = A^{(j=k)}$ and $e_k{}^\top A = A^{(i=k)}$ where e_k = 1 on k^{th} element and 0 everywhere else
- Matrix form:
  - In following $^{(j)}$ refers to column vector and $^{(i)}$ to row vector, however written as column vector
  - $u \cdot v = u^\top v = \sum_i u_i v_i = c$
  - $uv^\top = C$ with $u_i v_j = C_{ij}$
  - $Au = \sum_{j=i} A^{(j)} u_i = c$ with $A^{(i)} \cdot u = A^{(i)\top} u = c_i$
  - $u^\top A = \sum_{j=i} A^{(i)\top} u_j = c^\top$ with $u \cdot A^{(j)} = u^\top A^{(j)} = c_j$
  - $AB = \sum_{j=i} A^{(j)} B^{(i)\top} = C$ with $A^{(i)} \cdot B^{(j)} = A^{(i)\top} B^{(j)} = C_{ij}$
- Moving between instance-level $\rightarrow$ data-level:
  - $x^{(i)} y = a \rightarrow X^\top y = a$ where X consists of rows $x^{(i)}$
  - $x^{(i)} x^{(i)\top} = A \rightarrow X^\top X = A$ where X consists of rows $x^{(i)}$
  - $x^{(i)} \cdot \beta = y_i \rightarrow X\beta = y$ where X consists of rows $x^{(i)}$

Properties —

- $(A+B)^\top = A^\top + B^\top$
- $(\alpha A)^\top = \alpha A^\top$
- $(AB)^\top = B^\top A^\top$
- $(A+B)+C = A+(B+C)$
- $A+B = B+A$
- $\alpha(A+B) = \alpha A + \alpha B$
- $(\alpha+\beta)A = \alpha A + \beta A$
- $(\alpha\beta)A = \alpha(\beta A)$
- $(A+B)x = Ax + Bx = Cx$
- $(AB)x = A(Bx) = Cx$
- $A = 0.5(A + A^\top) + 0.5(A - A^\top) = B + C$ where B is symmetric, but not C
- $A = AI = IA$
- $Ak = AK = KA$
- $\text{rank}(AB) = \min(\text{rank}(A), \text{rank}(B))$
- $A^\top A$ satisfies:
  - Symmetric
  - Psd
  - Has rank m iff it is pd
  - Invertible iff it has rank m and it is pd
  - $\text{rank}(A^\top A) = \text{rank}(A) = \text{rank}(A^\top)$
  - $\text{rank}(A^\top A) = \text{rank}([A^\top A | A^\top x])$

Matrix terminology —

- Kernel $\text{null}(X)$ contains set of vectors b such that linear map $Xb = 0$
- Nullity = $\dim(\text{null}(X))$
- Image $\text{range}(X)$ contains set of vectors b generated by linear map Xb resp. is space spanned by columns of (X)
- Row space is space spanned by rows of (X)
- Column rank = $\dim(\text{colspace}(X))$ = number of linearly independent columns, row rank = $\dim(\text{rowspace}(X))$ = number of linearly independent rows
- Rank = column rank = row rank = $\dim(\text{range}(X)) = \dim(\text{range}(X^\top)) \leq min(n, m)$
- *Rank nullity theorem*: $\text{Rank}(X) + \text{nullity}(X) = m$

Matrices as linear maps — X maps b from $\mathbb{R}^m$ to $\mathbb{R}^n$: $Xb = y$ with $X \in \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^m$, $y \in \mathbb{R}^n$

- Injective: $Xb = y$ has at most one solution, happens iff columns of X are linearly independent ($\text{rank}(X) = m \leq n$)
- Surjective: $Xb = y$ has at least one solution, happens iff rows of X are linearly independent ($\text{rank}(X) = n \leq m$)
- Bijective: Mapping is both injective and surjective, i.e. $m = n$

Projection matrices —
Generally:

- Projection matrix satisfies $P = P^2$
- Proof:
  - Let S be spanned by $\{y_i\}_{i=1}^n$, which are column vectors of the matrix $A \in \mathbb{R}^{m \times n}$
  - Then, Ac are linear combinations of $\{y_i\}_{i=1}^n$
  - A vector $(x - Ac)$ is orthogonal to the columnspace of A, if: $\text{columnspace}(A) \cdot (x - Ac) = A^\top(x - Ac) = A^\top x - A^\top Ac = 0$
  - Then, $c = (A^\top A)^{-1} A^\top x$
  - With this definition of c, we have $A(A^\top A)^{-1} A^\top$ as the projection matrix P
  - $P^2 = (A(A^\top A)^{-1} A^\top)(A(A^\top A)^{-1} A^\top) = A(A^\top A)^{-1} A^\top = P$

Via orthonormal basis: Let S be spanned by orthonormal $\{b_i\}_{i=1}^n$, which are column vectors of the matrix $B \in \mathbb{R}^{m \times n}$

- Projection of x onto S is given by: $u = \sum_i (x \cdot b_i) b_i = \sum_i b_i b_i{}^\top x = BB^\top x = Cx$
- Projection of x onto $S^\perp$ is given by: $x - u = Ix - Cx$

Via SVD: Let S be spanned by $\{y_i\}_{i=1}^n$, which are column vectors of the matrix $A \in \mathbb{R}^{m \times n}$

- Projection of x onto S is given by: $s = AA^\# x$ since $AA^\#$ is a projection matrix due to $AA^\# = (AA^\#)^2$
- $s = U_+ U_+^\top x$ where U is obtained from SVD of A
- $\sum_{l=1}^m |u_k \cdot y_l| = \sigma_k^2$
  Proof: $\sum_{l=1}^m |u_k \cdot y_l| = ||u_k^\top A|| = u_k^\top AA^\top u_k = u_k^\top USV^\top VS^\top U^\top u_k = e_k^\top SS^\top e_k = \sigma_k^2$

Square Matrix Properties

Square matrix terminology —

- Diagonal matrix:
  - Def: Has $\{d_i\}_{i=1}^n$ on diagonal and 0 everywhere else
  - For diagonal matrices: $DD^\top = D^\top D$
- Inverse matrix:
  - Def: $A^{-1}A = I$
  - Is unique
  - For diagonal matrices: A^{-1} can be calculated by inverting all diagonal elements
- Symmetric (Hermitian) matrix:
  - $A^\top = A$
  - Properties:
    - $(x + A^{-1}b)^\top A(x + A^{-1}b) - b^\top A^{-1} b = x^\top Ax + 2x^\top b$
    - If A and B are symmetric, $A + B$ is also symmetric
- Orthogonal (unitary) matrix:
  - Def: $A^\top = A^{-1}$
  - $AA^\top = A^\top A = I$
  - Rows and columns are orthonormal
  - $||Ax|| = ||x||$
  - $(Ax) \cdot (Ay) = x \cdot y$
- Involution matrix: $A^{-1} = A$
- Determinant:
  - Function which maps A to a scalar
  - Properties:
    - $\det(I) = 1$
    - $\det(AB) = \det(A)\det(B)$
    - $\det(A^\top) = \det(A)$
    - $\det(A^{-1}) = (\det(A))^{-1}$
    - $\det(\alpha A) = \alpha^2 \det(A)$

Invertible matrix theorem — Following statements are equivalent for square matrix $A \in \mathbb{R}^{n\times n}$:

- A is invertible
- Only solution to $Ax = 0$ is $x = 0_v$
  Proof:
  - $A^{-1}Ax = 0 \Rightarrow Ix = 0 \Rightarrow x = 0_v$
- A is non-singular
- Columns (and rows) of A are linearly independent
- $\text{rank}(A) = n$
- $\det(A) = 0$

Inversely, if A is not invertible, the columns and rows are not linearly independent, etc.

Matrix inversion lemma —

- Let $B \in \mathbb{R}^{n\times n}, D \in \mathbb{R}^{m\times m}, C \in \mathbb{R}^{n\times m}$.
  Then, $A = B^{-1} + CD^{-1}C^\top$ is invertible:
  $A^{-1} = B - BC(D + C^\top BC)^{-1}C^\top B$
- Let $v \in \mathbb{R}^n$.
  Then, $(\alpha I + vv^\top)^{-1}v = (\alpha + \|v\|^2)^{-1}v = v^\top(\alpha I + vv^\top)^{-1} = v^\top(\alpha + \|v\|^2)^{-1}$

Quadratic form — Quadratic form of square matrix M: $x^\top Mx$. Can be expressed as quadratic form of a symmetric matrix A: $x^\top Ax$ where $A = 0.5\times(M + M^\top) + 0.5\times(M - M^\top)$.

Eigenvectors and eigenvalues —

- q is an eigenvector of A associated with an eigenvalue λ if it remains on the same line after transformation by a linear map: $Aq = \lambda q$
- Let $A \in \mathbb{R}^{n\times n}$. A can have between $1 - n$ eigenvalues, each with multiple eigenvectors. Eigenvectors for distinct eigenvalues are linearly independent.
- If there exists a non-trivial solution for q, $(A - \lambda I)$ is not invertible and characteristic polynomial $\det(A - \lambda I) = 0$
- *Eigendecomposition resp. diagonalization*: $A = Q\Lambda Q^{-1}$ where Q is a matrix with the eigenvectors as columns and Λ is a diagonal matrix with the eigenvalues on the diagonal
- $\det(A) = \det(Q\Lambda Q^{-1}) = \prod_{i=1}^n \lambda_i$
- *Symmetric eigendecomposition resp. unitary diagonalization*: For symmetric A: $A = Q\Lambda Q^\top$ where Q is an orthogonal matrix with the eigenvectors as columns and Λ is a diagonal matrix with the eigenvalues on the diagonal
- *Spectral theorem*: Square matrix A is symmetrically diagonizable, iff $AA^\top = A^\top A$
- *Spectral theorem for symmetric matrices*: Every symmetric matrix A is symmetrically diagonizable (due to Spectral theorem) and all its eigenvalues are real

Positive definite (pd) and positive semi-definite matrices (psd) —

- $A > 0$ iff $x^\top Ax > 0$
- $A \geq 0$ iff $x^\top Ax \geq 0$

Properties:

- If A is p(s)d, αA is also p(s)d
- If A and B are p(s)d, $A + B$ is also p(s)d
- If $\det(A) = \prod_{i=1}^n \lambda_i > (\geq)\ 0$ resp. $\{\lambda_i\}_{i=1}^n > (\geq)\ 0$ for pd (psd)

Pd properties:

- I is pd
- If A is pd, A^{-1} is pd
- *Cholesky decomposition*: If A is pd, $A = BB^\top$
- If A and B are pd, $(AB)^{-1} = B^{-1}A^{-1}$

Psd properties:

- If A is psd, $BAB^\top$ is psd

Singular Value Decomposition (SVD)

SVD —

- For $A \in \mathbb{R}^{n\times m}$, orthogonal rotation matrix $U \in \mathbb{R}^{n\times n}$, diagonal scaling and projection matrix $S \in \mathbb{R}^{n\times m}$, and orthogonal rotation matrix $V \in \mathbb{R}^{m\times m}$: $A = USV^\top$
- For symmetric $A \in \mathbb{R}^{n\times n}$: $A = USU^\top$
- In S:
  - Diagonal elements $\sigma_1, \ldots$ are the *singular values* of A
  - If $\sigma_1 \geq \sigma_2 \ldots \geq 0$, S is unique
  - *Condition number* = $\sigma_{max}/\sigma_{min}$
  - For square A: Iff $\sigma_1, \sigma_2, \ldots > 0$, A is invertible
- SVD is closely related to spectral theorem:
  - According to spectral theorem, every matrix A is symmetrically diagonizable (i.e. $A = Q\Lambda Q^\top$), iff $AA^\top = A^\top A$
  - If we apply SVD to $AA^\top$ resp. $A^\top A$:
    - $AA^\top = USV^\top VS^\top U^\top = U(SS^\top)U^\top$ since V is orthogonal and $V^\top V = I$
    - $A^\top A = VS^\top U^\top USV^\top = V(S^\top S)V^\top$ since U is orthogonal and $U^\top U = I$
  - $SS^\top$ and $S^\top S$ are diagonal matrices with elements $\sigma_1^2, \sigma_2^2, \ldots$
  - Given symmetric diagonalization for any matrix, we see that
    - S contains square root of eigenvalues of $AA^\top$ resp. $A^\top A$
    - U contains eigenvectors of $AA^\top$ as columns resp. V contains eigenvectors of $A^\top A$ as columns
  - According to spectral theorem, symmetric matrix A is symmetrically diagonizable (i.e. $A = Q\Lambda Q^\top$)
  - If we apply SVD to symmetric matrix A, we see that
    - S contains absolute value of eigenvalues of A
    - U contains eigenvectors of A as columns

Pseudo Inverse —

- Pseudo Inverse satisfies certain conditions that make it behave like an inverse for matrices that might not be invertible in the usual sense
- $A^\# = VS^\# U^\top$ where $S^\#$ is obtained from S by transposing it and taking the inverse of non-zero diagonal elements
- $A^\#$ is unique
- If $rank(A)$ = number of rows in A then:
  - $AA^\# = I$
  - $A^\# = A^\top(AA^\top)^{-1}$
  - Pseudo inverse provides mininum norm solution, when system $y = Ax$ is underdetermined: $x = A^\top(AA^\top)^{-1}y$
- If $rank(A)$ = number of columns in A then:
  - $A^\# A = I$
  - $A^\# = (A^\top A)^{-1}A^\top$
  - Pseudo inverse provides least squares solution, when system $y = Ax$ is overdetermined: $x = (A^\top A)^{-1}A^\top y$

Properties —

- $AA^\# A = A$
- $A^\# AA^\# = A^\#$
- $(A^\top)^\# = (A^\#)^\top$
- $(AA^\top)^\# = (A^\#)^\top A^\#$
- $A^\# x = 0 \Leftrightarrow x^\top A = 0 \Leftrightarrow A^\top x = 0$
- Properties can be proven by replacing A by its SVD and $A^\#$ by its definition
- Column space of $A^\#$ equals column space of $A^\top$
- Property can be proven by replacing A and $A^\#$ by their SVD

2 Calculus

Derivatives

Rules —

- Sum rule: $\frac{\partial f+g}{\partial x} = \frac{\partial f}{\partial x} + \frac{\partial g}{\partial x}$
- Product rule: $\frac{\partial f\times g}{\partial x} = f\times\frac{\partial g}{\partial x} + g\times\frac{\partial f}{\partial x}$
- Chain rule: $\frac{\partial f(g)}{\partial x} = \frac{\partial f}{\partial g}\times\frac{g}{\partial x}$

Common derivatives —

- $\frac{\partial x^n}{\partial x} = nx^{n-1}$
- $\frac{\partial log(x)}{\partial x} = \frac{1}{x}$
- $\frac{\partial e^{kx}}{\partial x} = k\times e^{kx}$
- $\frac{\partial\sqrt{x}}{\partial x} = \frac{1}{2\sqrt{x}}$

Partial and directional derivative —

- For a function that depends on n variables $\{x_i\}_{i=2}^n$, partial derivative is slope of tangent line along direction of one specific variable x_i
- Directional derivative is slope of tangent line along direction of selected unit vector u

Gradient —

- Given scalar-valued function $f: \mathbb{R}^n \to \mathbb{R}$, returns vector containing first-order partial derivatives: $\nabla_x f: [\frac{\partial f}{\partial x_1}\ldots\frac{\partial f}{\partial x_n}]^\top$
- Gradient points in direction of greatest upward slope of f
- Magnitude of gradient equals rate of change when moving into direction of greatest upward slope

Hessian —

- Given scalar-valued function $f: \mathbb{R}^n \to \mathbb{R}$, returns matrix containing second-order partial derivatives:

$$\mathcal{H} = \nabla_x^2 f: \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \ldots & \frac{\partial^2 f}{\partial x_1\partial x_n} \\ \ldots & \ldots & \ldots \\ \frac{\partial^2 f}{\partial x_n\partial x_1} & \ldots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

- $\mathcal{H}$ is symmetric

Jacobian —

- Given vector-valued function $f: \mathbb{R}^n \to \mathbb{R}^m$ with $f = [f_1(x), \ldots, f_m(x)]^\top$, returns matrix containing first-order partial derivatives:

$$\nabla_x f: \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \ldots & \frac{\partial f_1}{\partial x_n} \\ \ldots & \ldots & \ldots \\ \frac{\partial f_m}{\partial x_1} & \ldots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Matrix calculus rules —

- $\frac{\partial a^\top x}{\partial x} = a$
- $\frac{\partial x^\top Ax}{\partial x} = (A + A^\top)x$
- $\frac{\partial a^\top Ab}{\partial A} = ab^\top$
- For symmetric A: $\frac{\partial x^\top Ax}{\partial x} = 2Ax$
- For square A:
  - $\frac{\partial a^\top A^{-1}b}{\partial A} = (A^\top)^{-1}ac^\top(A^\top)^{-1}$
  - $\frac{\partial log(|A|)}{\partial A} = (A^\top)^{-1}$

Extrema

Conditions for local minima and maxima —

- Point is a stationary point, i.e. first-order derivative = 0
- If Hessian is pd, it's a local minimum, if Hessian is nd, it's a local maximum, if Hessian is indefinite, it's a saddle point
- Local minima and maxima are the unique global minima and maxima in strictly convex functions resp. one of possibly infinitely many global minima and maxima in convex functions

Convexity —

- For a convex function:
  - $f(\lambda x + (1-\lambda)y \leq \lambda f(x) + (1-\lambda)f(y)$ with $\lambda \in [0,1]$
  - Hessian of stationary point(s) is psd
  - Global minimum exists, but may not be unique
- For a strictly convex function:
  - $f(\lambda x + (1-\lambda)y < \lambda f(x) + (1-\lambda)f(y)$ with $\lambda \in [0,1]$
  - Hessian of stationary point is pd
  - Unique global minimum exists
- Sum of convex functions $f_2(x) + f_1(x)$ is also convex, sum of convex and strictly convex function is strictly convex
- Chain of convex functions $f_2(f_1(x))$, where outer function $f_2(x)$ is increasing, is also convex
- Scalar multiple of convex function $\lambda f(x)$, where $\lambda \geq 0$, is also convex
- Any norm is convex

Nature of optimum — What does Hessian and function look like?

- If Hessian is pd and loss function is strictly convex, stationary point is a global minimum, and there is a unique solution
- If Hessian is psd and loss function is convex, stationary point is a global minimum, and there may be a geometrically unique or infinitely many solutions
- If Hessian is p(s)d but loss function is not convex, stationary point may be a local minimum and there may be a geometrically unique or infinitely many solutions

Optimization approach — Is function differentiable, continuous, and are relevant terms invertible?

- If yes, analytically solvable
- If no, numerically solvable (e.g. via gradient descent)

Constrained optimization —

- Lagrangian function: $\mathcal{L}(x,\lambda) = f(x) + \lambda g(x)$, where $g(x)$ is an $(m-1)$ dimensional constraint surface and λ is the Lagrange multiplier
- $\nabla_x\mathcal{L} = \nabla_x f(x) + \lambda\nabla_x g(x)$
- $\nabla_\lambda\mathcal{L} = g(x)$
- Solution is feasible if it fulfills constraints and optimal, if no other feasible solution produces a lower error
- Minimizing over Lagrangian $\mathcal{L}(x,\lambda) = f(x) + \lambda g(x)$ corresponds to minimizing log-loss resp. negative likelihood:
  - $\hat{x} = argmax_{xp(D|x)\rho(x)}$
  - $= argmin_{x(-logp(D|x)+k(x))}$
  - where $k(x) = -log\rho(x)$

  $\mathcal{L}(x,\lambda) = f(x) + \lambda g(x)$

For equality constraints: Minimize $f(x)$ subject to $g(x) = 0$

- Gradient of $f(x)$ must be orthogonal to constraint surface, otherwise (if it points into any direction along the constraint surface) $f(x)$ could still decrease for movements along the constraint surface
- On the constraint surface, $g(x)$ is a constant, so moving along any direction on the constraint surface has a directional derivative of 0. Since the gradient of $g(x)$ points into the direction of steepest ascent, it must be orthogonal to the constraint surface, otherwise (if it points into any direction along the constraint surface) $g(x)$ would not be constant on the constraint surface
- Then, gradients are parallel at optimum: $\nabla_x f(x^*) = \lambda\times\nabla_x g(x^*)$
- To find x^* and λ^*:
  - $\nabla_x L = 0$, expresses parallelity condition at minimum x^*
  - $\nabla_\lambda L = 0$, expresses constraint
  - This is an unconstrained optimization problem
- Optimum x^* and λ^* represents a saddle point of $\mathcal{L}$

For inequality constraints: Minimize $f(x)$ subject to $g(x) \leq 0$

- If x^* lies in $g(x) < 0$, constraint is inactive
- Otherwise, if x^* lies in $g(x) = 0$, constraint is active:
  - Gradient of $f(x)$ must point towards $g(x) < 0$ region, otherwise (if it would point away from $g(x) < 0$ region) the optimum would lie in this region
  - Then, gradients are anti-parallel at optimum: $\nabla_x f(x^*) = -\lambda\times\nabla_x g(x^*)$
- To find x^* and λ^*:
  - $\nabla_x\mathcal{L} = 0$ subject to *Karush Kuhn Tucker conditions*:
    - $g(x) \leq 0$
    - $\lambda \geq 0$
    - *Complementary slackness condition*: $\lambda g(x) = 0$, with $\lambda = 0, g(x) < 0$ for inactive constraints and $\lambda > 0, g(x) = 0$ for active constraints
  - $\nabla_\lambda\mathcal{L} = 0$ given complementary slackness condition
  - This is not an unconstrained optimization problem, but can be solved via duality
- Optimum x^* and λ^* represents a saddle point of $\mathcal{L}$

For multiple constraints: Minimize $f(x)$ subject to m inequality constraints

$\{g^{(i)}(x) \leq 0\}_{i=1}^{m}$ and p equality constraints $\{h^{(j)}(x) = 0\}_{j=1}^{p}$

- Then, Lagrangian is given by: $\mathcal{L}(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \mu^{(i)} g^{(i)}(x) + \sum_{j=1}^{p} \lambda^{(j)} h^{(j)}(x)$
- Then, general solution x^*, λ^*, μ^* is given by: $\nabla_x \mathcal{L} = 0$ subject to:
  - $\{g^{(i)}(x) \leq 0\}_{i=1}^{m}$ and $\{h^{(j)}(x) = 0\}_{j=1}^{p}$
  - $\{\mu^{(i)} \geq 0\}_{i=1}^{m}$
  - $\{\mu^{(i)} g^{(i)}(x) = 0\}_{i=1}^{m}$

Primal problem:

- $\min_x [\max_{\lambda,\mu} \mathcal{L}]$
- $\max_{\lambda,\mu} \mathcal{L} = f(x) + \max_{\lambda,\mu} [\sum_{i=1}^{m} \mu^{(i)} g^{(i)}(x) + \sum_{j=1}^{p} \lambda^{(j)} h^{(j)}(x)]$
- Second term gives rise to barrier function:
  - $= 0$ subject to constraints being met, given complementary slackness condition for inequality constraints and $h^{(j)}(x) = 0$ for equality constraints, which implies that dual problem becomes $\min_x (f(x))$
  - $= \infty$ otherwise, which implies that primal problem cannot be solved

Solving inequality constraints via duality – *weak duality*:

- Weak duality always holds and gives a lower bound of minimum of primal problem
- Given minimax theorem, $\min_x [\max_{\lambda,\mu} \mathcal{L}] = f(x)$ (provided barrier function) $\geq \max_{\lambda,\mu} [\min_x \mathcal{L}]$
- $\min_x \mathcal{L}$ is an unconstrained optimization problem
- $\max_{\lambda,\mu} [\min_x \mathcal{L}]$ is a concave maximization problem

Solving inequality constraints via duality – *strong duality*:

- Strong duality holds under certain conditions, for example *Slater's condition* if there exists a solution that strictly fulfills all inequality constraints $\{g^{(i)}(x) < 0\}_{i=1}^{m}$
- Then, $\min_x [\max_{\lambda,\mu} \mathcal{L}] = \max_{\lambda,\mu} [\min_x \mathcal{L}]$
- $\min_x \mathcal{L}$ can be solved for general solution x^* in terms of λ, μ
- Plug x^* back into $\mathcal{L}$ and maximize to find solutions λ^*, μ^*
- Specify x^* based on λ^*, μ^*

3 Probability and Statistics

Terminology

Kolmogorov axioms — Probability space defined by:

- Sample space: All possible outcomes $\Omega = \{\omega_1, ..., \omega_n\}$
- Event space: All possible results, where an event is a subset of the sample space
- Probability measure: Function that assigns a probability to an event

Axioms:

- Event space must be a *sigma algebra*:
  - If A is in sample space, its complement is also in sample space
  - If $A_1, ... A_n$ are in sample space, their union is also in sample space
- Probability measure must satisfy:
  - $0 \leq \mathbb{P}(A) \leq 1$
  - $\mathbb{P}(\Omega) = 1$
  - If $A_1, A_2, ...$ are in sample space and do not intersect, then $\mathbb{P}(A_1 \cup A_2 \cup ...) = \int_{n=1}^{\infty} \mathbb{P}(A_n)$

Further properties:

- All sets than can be formed from left and right inclusive interval $[0, a]$ are events. On that basis: $(b, 1] = [0, b]^c \in$ event space.

Variables —

- Random variable:
  - Discrete random variable: Characterized by pmf
  - Continuous random variable: Characterized by pdf
- Independent random variables:
  - $\mathbb{P}(A|B) = \mathbb{P}(A)$ and $\mathbb{P}(B|A) = \mathbb{P}(B)$
  - $\mathbb{E}(AB) = \mathbb{E}(A)\mathbb{E}(B)$
  - Correlation is 0
  - $\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$
  - Functions of independent random variables are also independent
- Conditionally independent random variables: Two random variables $\mathcal{X}$ and $\mathcal{Y}$ are conditionally independent, if there is a confounder $\mathcal{L}$ that causally affects both variables, but if we control for this confounder, the variables are not causally connected
- I.I.D. random variables: Independent and from identical distribution

Events —

- Complement: $\mathbb{P}(A^C) = 1 - \mathbb{P}(A)$ and $\mathbb{P}(A \cup A^C) = \mathbb{P}(A)\mathbb{P}(A^C)$
- Disjoint / mutually exclusive vs. joint / mutually inclusive
- Subset $A \subset B$ with $\mathbb{P}(A) < \mathbb{P}(B)$

Probabilities —

- Marginal probability $\mathbb{P}(A)$: Probability for single variable: $p(\mathcal{X}) = \sum_{\mathcal{Y}} p(x, y)$ resp. $f(\mathcal{X}) = \int_{\mathcal{Y}} f(x, y) dy$
- Joint probability $\mathbb{P}(A \cap B)$: Probability for combination of variables, given by all possible combinations resp. convolution of their pdfs
- Conditional probability $\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$: Probability for variable, given other variable: $p(\mathcal{X}|\mathcal{Y}) = \frac{p(x,y)}{\sum_{\mathcal{X}} p(x,y)}$ resp. $f(\mathcal{X}|\mathcal{Y}) = \frac{f(x,y)}{\int_{\mathcal{X}} f(x,y) dy}$
  - $\mathbb{P}(A|B) = 1 - \mathbb{P}(A^C|B)$
  - $\mathbb{P}(A_1|B) + \mathbb{P}(A_2|B) + ... = 1$
  - If conditioning on subset S:

$$p(x|S) = \begin{cases} p(x)/p(x \in S) & x \in S \\ 0 & x \notin S \end{cases}$$

- Bayesian terminology:
  - Prior $\mathbb{P}(\text{parameter})$
  - Posterior $\mathbb{P}(\text{parameter}|\text{data})$
  - Likelihood $\mathbb{P}(\text{data}|\text{parameter})$
  - Evidence $\mathbb{P}(\text{data})$
- *Bayes theorem*: Posterior $\mathbb{P}(A|B) = \frac{\text{Likelihood } \mathbb{P}(B|A) \times \text{Prior } \mathbb{P}(A)}{\text{Evidence } \mathbb{P}(B)}$
- Attention! In $p(\cdot|\theta)$ the $|$ can either refer to parametrizing on θ (parameter is part of the distribution's form but isn't observed or fixed) or conditioning on θ (parameter takes a observed and fixed value, and we evaluate the distribution on this condition)

Measures

Expected value — $\mathbb{E}(\mathcal{X}) = \sum_{\mathcal{X}} x \times p(x)$ resp. $\mathbb{E}(\mathcal{X}) = \int_{-\infty}^{\infty} x \times f(x) dx$ with pmf resp. pdf —

Properties:

- $\mathbb{E}(\alpha) = \alpha$
- $\mathbb{E}(\alpha\mathcal{X} + \beta) = \alpha\mathbb{E}(\mathcal{X}) + \beta$
- $\mathbb{E}(\alpha\mathcal{X} + \beta\mathcal{Y}) = \alpha\mathbb{E}(\mathcal{X}) + \beta\mathbb{E}(\mathcal{Y})$
- For orthogonal variables: $\mathbb{E}((\mathcal{X} + \mathcal{Y})^2) = \mathbb{E}(\mathcal{X}^2) + \mathbb{E}(\mathcal{Y}^2)$
- For independent variables: $\mathbb{E}(\mathcal{X}\mathcal{Y}) = \mathbb{E}(\mathcal{X})\mathbb{E}(\mathcal{Y})$
- For vector $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$: $\mathbb{E}_y(\boldsymbol{A}\boldsymbol{x}) = \boldsymbol{A}\mathbb{E}_X(\boldsymbol{x})$

Cauchy Schwarz inequality: $\mathbb{E}(\mathcal{X}, \mathcal{Y})^2 \leq \mathbb{E}(\mathcal{X}^2)\mathbb{E}(\mathcal{Y}^2)$

Standard deviation — $\sqrt{\mathbb{V}(\mathcal{X})}$

Covariance —

- Univariate variance of a random variable: $\mathbb{V}(\mathcal{X}) = \mathbb{E}((\mathcal{X} - \mathbb{E}(\mathcal{X}))^2) = \mathbb{E}(\mathcal{X}^2) - \mathbb{E}(\mathcal{X})^2$ where $\mathbb{E}(\mathcal{X}^2)$ is the unnormalized correlation resp. inner product
- Univariate covariance of two random variables: $\text{Cov}(\mathcal{X}, \mathcal{Y}) = \mathbb{E}((\mathcal{X} - \mathbb{E}(\mathcal{X}))(\mathcal{Y} - \mathbb{E}(\mathcal{Y}))) = \mathbb{E}(\mathcal{X}\mathcal{Y}) - \mu_{\mathcal{X}}\mu_{\mathcal{Y}}$ where $\mathbb{E}(\mathcal{X}\mathcal{Y})$ is the unnormalized correlation resp. inner product
- Multivariate covariance matrix of a vector:
  - $\Sigma = \text{Cov}(\boldsymbol{\mathcal{X}}) = \mathbb{E}((\boldsymbol{\mathcal{X}} - \mathbb{E}(\boldsymbol{\mathcal{X}}))(\boldsymbol{\mathcal{X}} - \mathbb{E}(\boldsymbol{\mathcal{X}}))^\top) = \mathbb{E}(\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{X}}^\top) - \mathbb{E}(\boldsymbol{\mathcal{X}})\mathbb{E}(\boldsymbol{\mathcal{X}})^\top = \begin{bmatrix} \text{Var}(\mathcal{X}_1) & ... & \text{Cov}(\mathcal{X}_1, \mathcal{X}_m) \\ ... & ... & ... \\ \text{Cov}(\mathcal{X}_m, \mathcal{X}_1) & ... & \text{Var}(\mathcal{X}_m) \end{bmatrix}$ where $\boldsymbol{R} = \mathbb{E}(\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{X}}^\top)$ is the unnormalized correlation matrix
  - Σ and $\boldsymbol{R}$ are symmetric and psd
  - $\Sigma = \boldsymbol{R} - \boldsymbol{\mu}_X \boldsymbol{\mu}_X^\top$

Properties - variance:

- $\mathbb{V}(\alpha) = 0$
- $\mathbb{V}(\alpha\mathcal{X} + \beta) = \alpha^2\mathbb{V}(\mathcal{X})$
- $\mathbb{V}(\mathcal{X} + \mathcal{Y}) = \mathbb{V}(\mathcal{X}) + 2\text{Cov}(\mathcal{X}, \mathcal{Y}) + \mathbb{V}(\mathcal{Y})$
- For uncorrelated (and independent) variables: $\mathbb{V}(\mathcal{X} + \mathcal{Y}) = \mathbb{V}(\mathcal{X}) + \mathbb{V}(\mathcal{Y})$
- For independent variables: $\mathbb{V}(\mathcal{X}\mathcal{Y}) = \mathbb{E}((\mathcal{X}\mathcal{Y})^2)\mathbb{E}(\mathcal{X}\mathcal{Y})^2$
- For vector $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$: $\mathbb{V}_y = \boldsymbol{A}\mathbb{V}_X\boldsymbol{A}^\top$
- For zero-mean variable: $\mathbb{V}(\mathcal{X}) = \mathbb{E}(\mathcal{X}^2) - \mathbb{E}(\mathcal{X})^2 = \mathbb{E}(\mathcal{X}^2)$ since $\mathbb{E}(\mathcal{X}) = 0$

Properties - covariance:

- $\text{Cov}(\mathcal{X}, \mathcal{X}) = \mathbb{V}(\mathcal{X})$
- $\text{Cov}((\alpha\mathcal{X} + \beta\mathcal{Y}), \mathcal{Z}) = \alpha\text{Cov}(\mathcal{X}, \mathcal{Z}) + \beta\text{Cov}(\mathcal{Y}, \mathcal{Z})$
- If covariance of two random variables is 0, they are uncorrelated, but not necessarily independent. Then, $\mathbb{E}(\mathcal{X}\mathcal{Y}) = \mathbb{E}(\mathcal{X})\mathbb{E}(\mathcal{Y})$
- If covariance and unnormalized correlation of two random variables is 0, they are orthogonal, but not necessarily independent. Then, $\mathbb{E}(\mathcal{X}\mathcal{Y}) = 0$
- For vector $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$:
  - $\Sigma_y = \boldsymbol{A}\Sigma_X\boldsymbol{A}^\top$
  - $\boldsymbol{R}_y = \boldsymbol{A}\boldsymbol{R}_X\boldsymbol{A}^\top$
- For zero-mean variables: $\text{Cov}(\mathcal{X}, \mathcal{Y}) = \mathbb{E}(\mathcal{X}\mathcal{Y}) - \mu_{\mathcal{X}}\mu_{\mathcal{Y}} = \mathbb{E}(\mathcal{X}, \mathcal{Y})$ since $\mu_{\mathcal{X}} = \mu_{\mathcal{Y}} = 0$

Cauchy Schwarz inequality:

- $\text{Cov}(\mathcal{X}, \mathcal{Y})^2 \leq \mathbb{V}(\mathcal{X})\mathbb{V}(\mathcal{Y})$
- $\mathbb{E}(\mathcal{X}\mathcal{Y})^2 \leq \mathbb{E}(\mathcal{X}^2)\mathbb{V}(\mathcal{Y}^2)$

Correlation — Normalized covariance

- Univariate correlation of a random variable: $\text{Cor}(\mathcal{X}, \mathcal{Y}) = \frac{\text{Cov}(\mathcal{X}, \mathcal{Y})}{\sqrt{\mathbb{V}(\mathcal{X})}\sqrt{\mathbb{V}(\mathcal{Y})}}$
- Multivariate correlation matrix of a vector:
  - $\boldsymbol{P} = \text{Cor}(\boldsymbol{\mathcal{X}}) = \begin{bmatrix} 1 & ... & \text{Cor}(\mathcal{X}_1, \mathcal{X}_m) \\ ... & ... & ... \\ \text{Cor}(\mathcal{X}_m, \mathcal{X}_1) & ... & 1 \end{bmatrix}$
  - $\boldsymbol{P}$ is symmetric and psd
  - Correlation is bounded between 0 and 1, given Cauchy Schwarz Inequality
  - If correlation of two random variables is 0, they are not necessarily independent

Probability Distributions

PMF, CDF, PDF —

- Cumulative density function $F(r)$ (CDF): $F(r) = p(x \leq r)$
- Probability mass function $p(x)$ (PMF) for discrete random variables: $p(x)$
- Probability density function (PDF) $f(x)$ for continuous random variables: $\int_{-\infty}^{r} f(x) dx = p(x \leq r) = F(r)$
- Properties of CDF and PDF:
  - Derivative of CDF returns PDF, integral of PDF returns CDF
  - Monotonically non-decreasing: If $s < r, F(s) < F(r)$
  - $lim_{r \to -\infty} F(r) = 0$
  - $lim_{r \to \infty} F(r) = 1$
  - Right-continuous: $lim_{s \to -r^+} F(s) = F(r)$
  - $lim_{s \to -r^-} F(s) = F(x < r) = F(s) - F(x = r)$
  - $\int_a^b f(x) dx = F(b) - F(a) = p(a < x \leq b)$
  - $\int_{-\infty}^{\infty} f(x) dx = 1$

Normal distribution — $\mathcal{X} \sim \mathcal{N}(\mu, \sigma^2)$

For univariate, PDF: $\frac{1}{\sqrt{2\pi\sigma}} exp(\frac{-(x-\mu)^2}{2\sigma^2})$

For multivariate, PDF: $\frac{1}{2\pi\sigma^{n/2}} \frac{1}{|\Sigma|^{1/2}} exp(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^\top \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}))$ where the term in the exponent is a quadratic form

Convolution: $\int \mathcal{N}(a; Bc, D) \times \mathcal{N}(c; e, F) dc = \int \mathcal{N}(a; Be, D + BFB^\top$

Bernoulli distribution — trial with success (probability p) or failure (probability $1 - p$)

- $\mathcal{X} \sim \text{Bernoulli}(p)$
- PDF: $p(x) p^x (1 - p)^x$
- Mean: $\mathbb{E}(x) = p$
- Variance: $\mathbb{V}(x) = p(1 - p)$

Binomial distribution — n independent Bernoulli trials with k successes

- $\mathcal{X} \sim \text{Bin}(n, p)$
- PDF: $\binom{n}{k} p^k (1 - p)^{n-k}$
- Mean: $\mathbb{E}(x) = np$
- Variance: $\mathbb{V}(x) = np(1 - p)$

Poisson distribution —

- $\mathcal{X} \sim \text{Pois}(\lambda)$
- PDF: $e^{-\lambda} \frac{\lambda^x}{x!}$
- Mean: $\mathbb{E}(x) = \lambda$
- Variance: $\mathbb{V}(x) = \lambda$

Beta distribution —

- $\mathcal{X} \sim \text{Beta}(\alpha, \beta)$
- PDF: $\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)+\Gamma(\beta)} x^{\alpha-1} (1 - x)^{\beta-1}$
- Mean: $\mathbb{E}(x) = \frac{\alpha}{\alpha+\beta}$
- Variance: $\mathbb{V}(x) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$

Laws of Large Numbers and Inqequalities

Laws of large numbers — Sample mean of iid variables converges to population mean as $n \to \infty$

Jensen's inequality — Relates expected value of a convex function of a random variable to the convex function of the expected value of that random variable $\mathbb{E}(f(\mathcal{X})) \geq f(\mathbb{E}(\mathcal{X}))$

Markov's inequality — $p(x \geq t) \leq \frac{\mathbb{E}(x)}{t}$

Interesting only for $t \geq \mathbb{E}(x)$ because $p(x \geq t)$ must then be less than or equal to 1

Generalizations:

- $p(|x| \geq t) \leq \frac{\mathbb{E}(g(|x|))}{g(t)}$
- $p(|x| \geq t) \leq \frac{\mathbb{E}(|x|^n)}{t^n}$

Chebychev's inequality — $p(|x - \mu_x| \geq \alpha|\sigma_x|) \leq \frac{1}{\alpha^2}$

Interesting only for $\alpha > 1$

Implications:

- For n variables: $p(|S_n - \mu_x| \geq \epsilon) \leq \frac{\sigma_x^2}{n\epsilon^2}$ where S_n is the sample mean

4 Information Theory

Description

Entropy —

- $H(x) = -\sum_x p(x) log(p(x)) = -\sum_{x,y} p(x, y) log(p(x))$ resp. $H(x) = -\int p(x) log(p(x)) dx$
- Measure of randomness in a variable resp. quantifies uncertainty of a distribution

Properties:

- $H(x) \geq 0$
- $H(x)$ is maximized, when x is a uniform random variable
- For independent variables: $H(x, y) = H(x) + H(y)$

Conditional entropy —

- $H(x|y) = -\sum_{x,y} p(y) p(x|y) log(p(x|y)) = -\sum_{x,y} p(x, y) log(\frac{p(x,y)}{p(y)})$
- Measure of how much information of x is revealed by y

Properties:

- $0 \leq H(x|y) \leq H(x)$ with equality if when x is independent with y resp. if y completely determines x

Mutual information —

- $I(x; y) = H(x) - H(x|y) = -\sum_{x,y} p(x, y) log(\frac{p(x)p(y)}{p(x,y)})$
- Measure of how much information of x is left after y is revealed

Properties:

- $0 \leq I(x; y) \leq H(x)$ with equality if y completely determines x resp. if x is independent with y

KL divergence —

- $KL(p; q) = \sum_x p(x) log(\frac{p(x)}{q(x)})$

- Measures the extra information or inefficiency when approximating a true distribution over x, p, with a predicted one, q

Properties:

- $KL(p;q) \geq 0$

Cross entropy —

- $CE(p|q) = KL(p;q) + H(p) = -\sum_x p(x) log(q(x))$
- Measures the total uncertainty when using the predicted distribution q to represent the true distribution p, combining both the model's error and the intrinsic uncertainty of the true distribution

Properties:

- $KL(p;q) \geq 0$

5 ML Paradigms

Frequentism

Description —

- Parametric approach
- θ as fixed, unknown quantity, $\boldsymbol{X}$ as random, and known quantity
- Makes point estimate
- Focuses on maximizing likelihood $p(\boldsymbol{X}|\theta)$ to infer posterior $p(\theta|\boldsymbol{X})$
- Only requires differentiation methods
- High variance, but low bias

MLE estimator

- Maximizes log-likelihood: $\hat{\theta} = argmax_{\boldsymbol{\theta}}(L) = argmax_{\boldsymbol{\theta}}(\prod_{i=1}^{n} p(\boldsymbol{x_i}|\boldsymbol{\theta})) = argmax_{\boldsymbol{\theta}}(\sum_{i=1}^{n} log(p(\boldsymbol{x_i}|\boldsymbol{\theta})))$
- Advantages
  - Consistent: $\hat{\theta} \rightarrow \theta$ as $n \rightarrow \infty$
  - Asymptotically normal: $\frac{1}{\sqrt{n}}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})$ coverges to $\mathcal{N}(0, \boldsymbol{J}^{-1}(\boldsymbol{\theta})\boldsymbol{I}(\boldsymbol{\theta})\boldsymbol{J}^{-1}(\boldsymbol{\theta}))$ where $\boldsymbol{J} = -\mathbb{E}[\frac{\partial^2 log(p(\boldsymbol{x}|\boldsymbol{\theta}))}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^\top}]$ and where $\boldsymbol{I}$ is the Fisher information
  - Asymptotically efficient: $\hat{\theta}$ minimizes $\mathbb{E}[(\hat{\theta} - \theta)^2]$ as $n \rightarrow \infty$
    - Not necessarily the best estimator, especially for small samples in a multivariate context
    - (cf. Rao-Cramer bound)
  - Equivariant: If $\hat{\theta}$ is MLE of θ, then $g(\hat{\theta})$ is MLE of $g(\theta)$
- Proofs of advantages
  - Asymptotically normal:
    - We start with the score and set it to 0 for optimization with regard to θ: $\Lambda = \frac{\partial}{\partial\boldsymbol{\theta}} log(p(\boldsymbol{x}|\boldsymbol{\theta})) = 0$
    - With a Taylor expansion, we can show that $(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})\sqrt{n} = \frac{1}{\sqrt{n}}\Lambda[-\frac{1}{n}\frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^\top}\sum_{i=1}^{n} log(p(x_i|\boldsymbol{\theta}))]^{-1}$ where Λ is the score
    - We set $\boldsymbol{J} = [-\frac{1}{n}\frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^\top}\sum_{i=1}^{n} log(p(x_i|\boldsymbol{\theta}))]$
    - $\frac{1}{\sqrt{n}}\Lambda$ is a random vector with covariance matrix $\boldsymbol{I}$ and converges to the normal distribution $\sim \mathcal{N}(0, \boldsymbol{I})$
    - Then, $(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})\sqrt{n} = \boldsymbol{J}^{-1}\frac{1}{\sqrt{n}}\Lambda \sim \boldsymbol{J}^{-1}\mathcal{N}(0, \boldsymbol{I})$
    - $\mathbb{V}(\boldsymbol{J}^{-1}\frac{1}{\sqrt{n}}\Lambda) = \mathbb{E}[\boldsymbol{J}^{-1}\boldsymbol{I}\boldsymbol{J}^{-1}]$
    - This equality is given because $\mathbb{V}(x) = \mathbb{E}[x - \mathbb{E}(x)] = \mathbb{E}[x]$ if $\mathbb{E}(x) = 0$, which is the case here, given that the expected score is 0
    - So we have shown that $\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})\sqrt{n} = \boldsymbol{J}^{-1}\frac{1}{\sqrt{n}}\Lambda \sim \mathcal{N}(0, \boldsymbol{J}^{-1}\boldsymbol{I}\boldsymbol{J}^{-1})$
  - Equivariant:
    - Let $t = g(\boldsymbol{\theta})$ and $h = g^{-1}$
    - Then, $\boldsymbol{\theta} = h(t) = h(g(\boldsymbol{\theta}))$
    - For all t we have: $L(t) = \prod_i p(\boldsymbol{x}^(\boldsymbol{i})|h(t)) = p(\boldsymbol{x}^(\boldsymbol{i})|\boldsymbol{\theta}) = L(\boldsymbol{\theta})$
    - Hence, for all t we can say: $L(t) = L(\boldsymbol{\theta})$ and $L(\hat{t}) = L(\hat{\boldsymbol{\theta}})$

PAC estimator

- Generates probabilistic bounds for parameter θ that is approximately known with a high probability:
  - Probability of being correct: $1 - \delta$
  - Degree of approximation: ϵ
- Given Hoeffding's inequality, the probability that the error is greater than ϵ is bounded

Bayesianism

Description —

- Parametric approach
- θ as random, unknown quantity, $\boldsymbol{X}$ as random, and known quantity
- Makes estimate in form of distribution
- Leverages prior and likelihood to infer posterior: $p(\theta|\boldsymbol{X}, \boldsymbol{y}) = \frac{p(\theta)p(\boldsymbol{y}|\boldsymbol{X},\theta)}{p(\boldsymbol{y}|\boldsymbol{X})} = \frac{p(\theta)p(\boldsymbol{y}|\boldsymbol{X},\theta)}{\int p(\theta)p(\boldsymbol{y}|\boldsymbol{X},\theta)d\theta} \propto p(\theta)p(\boldsymbol{y}|\boldsymbol{X},\theta) = p(\theta, \boldsymbol{y}|\boldsymbol{X})$
- Focuses on minimizing cost function $\mathbb{E}[k(\boldsymbol{\theta}', \boldsymbol{\Theta})|\boldsymbol{X}, \boldsymbol{y}] = \int_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{y}) \times k(\boldsymbol{\theta}', \boldsymbol{\theta})d\boldsymbol{\theta} \propto \int_{\boldsymbol{\theta}} p(\boldsymbol{\theta}, \boldsymbol{y}|\boldsymbol{X}) \times k(\boldsymbol{\theta}', \boldsymbol{\theta})d\boldsymbol{\theta}$
- Requires integration methods for normalizing constant in denominator, which can be intractable, in which case MAP estimator can provide an alternative
- Low variance, but high bias

Mean estimator

- Minimizes mean squared error as cost function $k(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) = |\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}|^2$
- The resulting estimate is the mean of the posterior: $\hat{\boldsymbol{\theta}} = \mathbb{E}[\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{y}]$
- Returns estimate that reflects central tendency and overall uncertainty

MAP estimator

- Generally, maximizes posterior: $\hat{\boldsymbol{\theta}} = argmax_{\boldsymbol{\theta}}(p(\boldsymbol{\theta}|\boldsymbol{X}))$ resp. $p(\boldsymbol{\theta}|\boldsymbol{X})p(\boldsymbol{X})$
- In discrete cases, minimizes following cost function:

$$k(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta} = \begin{cases} 1 & \hat{\boldsymbol{\theta}} \neq \boldsymbol{\theta} \\ 0 & \hat{\boldsymbol{\theta}} = \boldsymbol{\theta} \end{cases}$$

- Returns single point estimate

Statistical Learning

Description —

- We want to minimize expected risk $\mathcal{R}(f) = \mathbb{E}_{X,Y}[1f(X) \neq Y]$, but this is difficult because
  - We don't have access to the joint distribution of X, Y
  - We cannot find f, without any assumptions on its structure
  - It's unclear how to minimize the expected value
- Therefore, we make following choices:
  - We collect sample Z
  - We restrict space of possible choices of f to a set $\mathcal{H}$
  - We use a loss function to approximate the expected value
- With these choices, we approximate the expected risk via the empirical risk $\hat{\mathcal{R}}(f) = \hat{L}(Z, f) = \frac{1}{n}\sum_i L(y_i, f(x_i))$

6 Model Taxonomy

Active Learning

Active learning —

- Assume:
  - Domain space $\mathcal{X}$
  - Sample space $S \subseteq \mathcal{X}$
  - Labeled data $D_{n-1}(x_i, y_i)_{i<n}$
  - Target space $\mathcal{A} \subseteq \mathcal{X}$
  - We estimate $y_x = f_x + \epsilon_x$
- We aim to find the next x_n that gives us the most information about f in $\mathcal{A}$
- Information gain can be quantified as maximizing the conditional mutual information between y_x and f: $IG[f_x; y_x|D_{n-1}] = H(y_x|D_{n-1}) - H(y_x|f_x, D_{n-1})$ where $H(y_x|D_{n-1})$ is the uncertainty about y_x before labeling x_n and $H(y_x|f_x, D_{n-1})$ is the uncertainty about y_x after labeling x_n. We want to minimize the latter, i.e. we want to maximize the delta between the former and the latter
- We pick $x_n = argmax_{x \in S} IG[f_x; y_x|D_{n-1}]$
- To find a closed-form solution, we assume that f is a Gaussian process with a known mean and kernel function:
  - $f \sim \mathcal{GP}(\mu, k)$
  - $\boldsymbol{f} = (f_{x_1}, f_{x_2}, ...) \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ where elements in mean vector are $\mu_i = \mu(x_i)$ and elements in covariance matrix are $\Sigma_{ij} = k(x_i, x_j)$
- Under this assumption, we can show that $IG[f_x; y_x|D_{n-1}] = \frac{1}{2}log(\frac{\mathbb{V}(y_x|D_{n-1})}{\mathbb{V}(y_x|f_x, D_{n-1})})$

  Proof:
  - Gaussian entropy of $a|B$ given by: $H(a|B) = -\int p(a|B) log(p(a|B))da = -\mathbb{E}[log\mathcal{N}(\mu, \sigma)] = -\mathbb{E}[log((2\pi\sigma^2)^{-1/2}exp(-\frac{(x-\mu)^2}{2\sigma^2}))] = \frac{1}{2}log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\mathbb{E}[(x-\mu)^2] = \frac{1}{2}log(2\pi\sigma^2) + \frac{1}{2} \times 1 = \frac{1}{2}log(2\pi\sigma^2) + \frac{1}{2}log(e) = \frac{1}{2}log(2\pi e\sigma^2)$
  - Plugging this in, we get: $H(y_x|D_{n-1}) - H(y_x|f_x, D_{n-1}) = \frac{1}{2}log(2\pi e\mathbb{V}(y_x|D_{n-1})) - \frac{1}{2}log(2\pi e\mathbb{V}(y_x|f_x, D_{n-1})) = \frac{1}{2}log(\frac{\mathbb{V}(y_x|D_{n-1})}{\mathbb{V}(y_x|f_x, D_{n-1})})$

Safe Bayesian learning —

- Bayesian approach to active learning
- Assume:
  - We have stochastic process f^*
  - We can iteratively choose points $x_1, ..., x_{n-1} \in \mathcal{X}$ and observe $y_i = f^*(x_1), ..., y_{n-1} = f^*(x_{n-1})$
  - Points should lie in safe area S^* which is the set of $x \in \mathcal{X}$ such that another stochastic process $g^*(x) \geq 0$
  - For chosen points, we can also observe $z_i = g^*(x_1), ..., z_{n-1} = g^*(x_{n-1})$ which are measurements of confidence, indicating high confidence when above 0
- We aim to find estimates of sample space S and target space $\mathcal{A}$
- To do so, we fit a Gaussian process on observed $\{(x_i, y_i)\}_{i<n}$ and $\{(x_i, z_i)\}_{i<n}$. Gaussian process over f and g induces two bounds respectively, which provide the 95% confidence interval of $\mathbb{E}[f(x)]$ resp. $\mathbb{E}[g(x)]$:
  - Upper bound function $u_n^f(x)$ resp. $u_n^g(x)$
  - Lower bound function $l_n^f(x)$ resp. $l_n^g(x)$
- Gaussian process over g allows to derive pessimistic and optimistic estimate of safe area:
  - Pessimistic: $S_n = \{x : l_n^g(x) \geq 0\}$
  - Optimistic: $\hat{S}_n = \{x : u_n^g(x) \geq 0\}$
- We then gather estimates, where upper bound of f lies above baseline set by maximum value of lower bound of f: $\mathcal{A}_n = \{x \in \hat{S}_n : u_n^f(x) \geq max_{x' \in S_n} l_n^f(x')\}$
- We can then perform active learning with sample space $S = S_n$ and target space $\mathcal{A} = \mathcal{A}_n$

Batch active learning —

- Variant of active learning
- Assume:
  - Domain space $\mathcal{X}$ and distribution P over $\mathcal{X}$
  - Oracle to unknown function $f : \mathcal{X} \rightarrow \mathcal{Y}$
  - Population set $\mathcal{X} = \{x_1, ..., x_m\} \subseteq \mathcal{X}$
  - Budget $b \leq m$
- We aim to find next batch of data points $L \subseteq \mathcal{X}$ subject to $|L| = b$ that gives us the most information
- Suppose we know $Z = \{(x, f(x)) : x \in L\}$
- 1-nearest-neighbor classifier $\hat{f}$ is fitted to Z
- Let $B_\delta(x) = \{x' \in \mathcal{X} : \|x - x'\| \leq \delta\}$ be the set of sufficiently close points to x
  - We consider $B_\delta(x)$ pure if f yields same results for all of $B_\delta(x)$
- Impurity of δ is given by $\hat{\pi}(\delta) = P(\{x \in \mathcal{X} : B_\delta(x) \text{ is not pure}\})$
- Let $C(L, S) = \bigcup_{x \in L} B_\delta(x)$ be the union of all sets B
  - $C = C_r \cup C_w = \{x \in C : \hat{f}(x) = f(x)\} \cup \{x \in C : \hat{f}(x) \neq f(x)\}$
- We have $C_w \subseteq \{x \in \mathcal{X} : B_\delta(x) \text{ is not pure}\}$. Then, $P(C_w) \leq \hat{\pi}(\delta)$
- $\{x : \hat{f}(x) \neq f(x)\} \subseteq C_w \cup C_r^C \subseteq \{x \in \mathcal{X} : B_\delta(x) \text{ is not pure}\} \cup C^C$
- Then, we have $\mathcal{R}(\hat{f}) = P(\hat{f}(x) \neq f(x) \leq \hat{\pi}(\delta) + 1 - P(C)$
- We need to choose L and δ such that $\mathcal{R}(\hat{f})$ is minimized
- We approach this by minimizing the upper bound, by picking δ and choosing C that maximizes $P(C)$: $argmax_{L \subseteq \mathcal{X}, |L|=b} P(\bigcup_{x \in L} B_\delta(x))$
- Two challenges:
  1. We don't know the distribution
  2. Problem is NP-hard
- We address 1) by using the empirical distribution induced by X. Then, we have: $argmax_{L \subseteq \mathcal{X}, |L|=b} \frac{1}{|X|}|\{x' : \|x' - x\| \leq \delta, \text{ for some } x \in L\}|$
- We address 2) with greedy algorithm:
  - Input: $x \subseteq \mathcal{X}, b \in \mathbb{N}$
  - Output: $L \subseteq X$ of size b
  1. $G = (x, E)$ where $E = \{(x, x') : \|x - x'\| \leq \delta\}$
  2. $L = \varnothing$
  3. For $i = 1, ..., b$:
     (a) $\hat{x} \leftarrow argmax_{x \in \mathcal{X}}|\{x' : (x, x') \in E, x \in \mathcal{X}\}|$
     (b) $L \leftarrow L \cup \hat{x}$
     (c) $E \leftarrow E - (\{\hat{x}\} \times (B_\delta(\hat{x}) \cap x))$
  4. Return L

Ensemble Methods

Motivation —

- Let $\hat{f}_1(x), ..., \hat{f}_B(x)$ be estimators
- When averaging estimators...
  - Average remains unbiased, if all estimators are unbiased

    Proof:

    Bias $= \mathbb{E}[\hat{f}(x)] - \mathbb{E}[y|x] = \frac{1}{B}\sum_{i=1}^{B}\mathbb{E}[\hat{f}_i(x)] - \mathbb{E}[y|x] = \frac{1}{B}\sum_{i=1}^{B} \text{bias}(\hat{f}_i(x))$
  - Variance is reduced by a factor of $\frac{1}{B}$, if the estimators have similar variance and no covariance

    Proof:
    - Variance $= \mathbb{E}[\hat{f}(x) - \mathbb{E}[\hat{f}(x)]]^2 = \mathbb{E}[\frac{1}{B}\sum_{i=1}^{B}\hat{f}_i(x) - \frac{1}{B}\sum_{i=1}^{B}\mathbb{E}[\hat{f}_i(x)]]^2 = \mathbb{E}[\frac{1}{B}\sum_{i=1}^{B}(\hat{f}_i(x) - \mathbb{E}[\hat{f}_i(x)])]^2 = \frac{1}{B^2}\sum_{i=1}^{B}\mathbb{V}[\hat{f}_i(x)] + \frac{1}{B^2}\sum\sum_{i \neq j=1}^{B} \text{Cov}[\hat{f}_i(x), \hat{f}_j(x)]$
    - Assuming variance are similar ($\approx \sigma^2$) and covariances are small (≈ 0), we get: Variance $= \frac{1}{B^2}\sum_{i=1}^{B}\sigma^2 = \frac{1}{B^2}B\sigma^2 = \frac{\sigma^2}{B}$

Requirements — Diversity of estimators, to reduce covariance. Achieved by:

- Different subsets of data for each estimator, e.g. via bootstrapping
- Different features for each estimators
- Decorrelating estimators during training

Variants —

- Regression: Average output of all estimators: $\hat{r}_B(x) = \frac{1}{B}\sum_{b=1}^{B} r_b(x))$
- Classification: Majority or weighted voting: $\hat{c}_B(x) = sgn(\sum_{b=1}^{B}\alpha_b c_b(x))$ with majority voting if $\alpha = 1$

Bootstrap Aggregating (Bagging) —

- Algorithm
  1. For $b = 1, ..., B$:
     (a) Hold out $\frac{1}{3}$ of sample

    (b) Construct $Z_b^* = b^{th}$ bootstrap sample from remaining $\frac{2}{3}$ of sample
    (c) Construct estimator f_b based on Z_b^*
2. Return $\hat{f}_B(x) =$ weighted average of $f_1(x), ..., f_B(x)$
3. Calculate out-of-bag error on held out sample

If desired: Estimators can be constructed in multiple function classes $f^{'}, f^{''}, ...$ and set of estimators in the function class, which generates the lowest empirical error, is returned

Random Forest —

- Algorithm:
  1. Generate multiple training sets via bootstrapping
  2. Construct multiple decision trees based on the generated training sets, where each tree selects a random set of features at each split via bootstrapping
     - Classification: Choose $m = \sqrt{k}$ predictors at each split
     - Regression: Choose $m = \frac{k}{3}$ predictors at each split
  3. Generate prediction by averaging or voting on trees
  4. Calculate out-of-bag error
- 2 sources of randomness:
  - Each tree trained on bootstrap sample of instances
  - At each split, bootstrap sample of features is considered

AdaBoost — Algorithm:

1. Each instance weight is initially $w_i = \frac{1}{n}$
2. Train first classifier $\hat{c}^{(1)}$ generating output $\hat{y}^{(1)}$
3. Calculate weighted error rate of j^{th} classifier: $r^{(j)} = \frac{\sum_{i=1}^n w_i \mathbb{I}_{\{\hat{y}_i^{(j)} \neq y_i\}}}{\sum_{i=1}^n w_i}$
4. Calculate classifier weight, which is higher, if the classifier has a lower error rate: $\alpha^{(j)} = \eta log(\frac{1-r^{(j)}}{r^{(j)}})$
5. Update instance weights: For $i = 1, ..., n$:
$$w_i = \begin{cases} w_i & \hat{y}_i^{(j)} = y_i \\ w_i e^{\alpha^{(j)}} & \hat{y}_i^{(j)} \neq y_i \end{cases}$$
6. Normalize instance weights: $w_i = \frac{w_i}{\sum_{i=1}^n w_i}$
7. Continue by training next classifier
8. ...
9. Generate prediction: $\hat{y}(x) = argmax_k \sum_{j=1}^B \alpha_j \mathbb{I}_{\{\hat{c}^{(j)}(x)=k\}}$

AdaBoost is an additive logistic model that minimizes the exponential loss function:

- Exponential loss function: $\mathbb{E}[e^{-yF(x)}] = p(y=1|x)e^{-F(x)} + p(y=-1|x)e^{F(x)}$
- Minimizer of exponential loss function is the log-odds: $\mathbb{E}[e^{-yF(x)}]$ is minimized at $\frac{\partial \mathbb{E}[e^{-yF(x)}]}{\partial F(x)} = \frac{1}{2} log(\frac{p(y=1|x)}{p(y=-1|x)}) = F(x) = 0$
- Then, $p(y=1|x) = \frac{e^{F(x)}}{e^{-F(x)} + e^{F(x)}}$
- Log-odds minimizer corresponds to the AdaBoost minimizer, since AdaBoost models the final output as a sum of classifiers: $log(\frac{p(y=1|x)}{p(y=-1|x)}) = \sum_{j=1}^B \alpha_j \hat{c}^{(j)}(x) = F(x)$

AdaBoost as gradient descent:

- Discrete AdaBoost model can be viewed as performing gradient descent on an additive logistic model
- Cost function: Minimize $J(F) = \mathbb{E}[e^{-yF(x)}]$ where $F(x)$ is the combined classifier, consisting of multiple weak classifiers
- Suppose we consider an improved function $F'(x)$: $F'(x) = F(x) + \alpha c(x)$ where $c(x)$ is a new weak classifier
- To approximate $J(F')$ we can perform a Taylor expansion around 0: $J(F') = \mathbb{E}[e^{-y(F(x)+\alpha c(x))}] = \mathbb{E}[e^{-yF(x)}(1 - y\alpha c(x) + \frac{1}{2}\alpha^2)] + O(\alpha^3)$
- To minimize $J(F')$ wrt $c(x)$:
  - We define weighted expectation:
    - Weights: $w(x,y) = e^{-yF(x)}$
    - Weighted expectation of $g(x,y)$: $\mathbb{E}[g(x,y)|x] = \frac{\mathbb{E}[w(x,y)g(x,y)|x]}{\mathbb{E}[w(x,y)|x]}$
  - We aim to choose $c(x)$ that maximizes weighted expectation of $g(x,y) = yc(x)$ since this is equivalent to minimizing exponential loss
  - In $O(\alpha^2)$ this yields: $c(x) =$
    - $argmin_c = \mathbb{E}[1 - y\alpha c(x) + \frac{1}{2}\alpha^2|x]$
    - $argmax_c = \mathbb{E}[yc(x)|x]$
    - $= 1$ if $\mathbb{E}[y|x] = 1 \times p(y = 1|x) + (-1) \times p(y = -1|x) > 0$ $= -1$ otherwise
  - We can approximate the exponential via the quadratic loss. Then, we have: $\frac{1}{2}\mathbb{E}[(y - c(x))^2] - 1 = \mathbb{E}[y^2 - 2yc(x) + c(x)^2]\frac{1}{2} - 1 = \mathbb{E}[1 - 2yc(x) + 1]\frac{1}{2} - 1 = -\mathbb{E}[yc(x)]$
  - Then, we have:
    - $argmin_c = -\mathbb{E}[yc(x)|x]$
    - $argmax_c = \mathbb{E}[yc(x)|x]$
  - Minimizing the quadratic approximation leads to a Newton-like step for choosing $c(x)$, making it a weighted least squares choice of $c(x)$
- To minimize $J(F')$ wrt α:
  - $\alpha^* = argmin_\alpha \mathbb{E}[e^{-y\alpha c(x)}] = e^\alpha \mathbb{E}[\mathbb{I}_{\{y \neq c(x)\}}] + e^{-\alpha}\mathbb{E}[\mathbb{I}_{\{y = c(x)\}}] = \frac{1}{2} log(\frac{1-err}{err})$ where $err = \mathbb{E}[\mathbb{I}_{\{y \neq c(x)\}}]$
- Combination yields AdaBoost update:
  - $F'(x) \leftarrow F(x) + \alpha^* c(x) = F(x) + \frac{1}{2} log(\frac{1-err}{err})$
  - $w(x,y) \leftarrow w(x,y) \times e^{-\alpha^* yc(x)} = w(x,y) \times e^{\alpha^*(2\mathbb{I}_{\{y \neq c(x)\}} - 1)} = w(x,y) \times e^{log(\frac{1-err}{err})\mathbb{I}_{\{y \neq c(x)\}}} \times e^{-\alpha^*}$ where $e^{-\alpha^*}$ is a constant

7 Model Optimization

Gradient Descent

Numeric optimization procedure

Gradient descent —

- Uses entire training set to evaluate whether new parameter is more optimal than previous one
- Slow and less likely to escape local minima due to randomness, but accurate
- Algorithm:
  1. Set $\eta > 0$
  2. Randomly initialize $\beta_{(t=0)}$
  3. $\beta_{(t+1)} \leftarrow \beta_{(t)} - \eta \nabla_\beta LO|_{\beta=\beta_{(t)}}$
  4. $t \leftarrow t + 1$
  5. Repeat 3 and 4 until $\nabla_\beta LO = 0$

Stochastic gradient descent —

- Uses only one training sample or mini-batch to evaluate whether new parameter is more optimal than previous one
- Fast and more likely to escape local minima due to randomness, but represents an approximation
- Algorithm:
  1. Set $\eta > 0$
  2. Randomly initialize $\beta_{(t=0)}$
  3. Shuffle training data and initialize $i \leftarrow 1$
  4. $\beta_{(t+1)} \leftarrow \beta_{(t)} - \eta \nabla_\beta LO \text{ for observation i } |_{\beta=\beta_{(t)}}$
  5. $t \leftarrow t + 1$
  6. $i \leftarrow i + 1$
  7. Repeat 4 to 6 until $i = n + 1$
  8. Repeat 2 to 6 until $\nabla_\beta LO = 0$
- Justification for SGD is given by *Robbins-Monro algorithm* which iteratively find the root (or zero) of an unknown function when only noisy observations of the function are available:
  - Algorithm:
    1. Choose learning rates $\eta_1, \eta_2, ...,$ typically decreasing over time
    2. Randomly initialize $\beta_{(t=0)}$
    3. $\beta_{(t+1)} \leftarrow \beta_{(t)} - \eta \nabla_\beta LO|_{\beta=\beta_{(t)}}$ where LO is noisy
  - For convergence:
    - $\sum_{t=1}^\infty \eta_t = \infty$ to ensure sufficient exploration
    - $\sum_{t=1}^\infty \eta_t^2 \leq \infty$ to avoid overly large updates
    - Then, $lim_{t \to \infty} p(|y_t - y| > \epsilon) = 0$ for any $\epsilon > 0$

Hyperparameters —

- Learning rate η: Determines step size, if too small algorithm is slow to converge, if too large algorithm may diverge
- Batch size b: Number of samples from training set used to evaluate optimality of β at each step
- Epoch: Number of times model works through entire training set. Every epoch, β is updated n/b times

Modifications —

- Data should be standardized resp. scaled, otherwise the gradient of the largest predictor dominates the gradient of the loss function, leading to uneven updating of β and slow convergence
- A momentum term can be added to the updating function to ensure smooth updating of β: $\beta_{(t+1)} \leftarrow \beta_{(t)} - \eta \nabla_\beta LO|_{\beta=\beta_{(t)}} + \alpha(\beta_{(t)} - \beta_{(t)-1})$
- For stochastic gradient descent, a smoothing step can be added because stochastic gradient descent hovers around desired solution: $\hat{\beta}_{(t+1)} \leftarrow \frac{1}{L+1}\sum_{j=t-L}^t \beta_{(t)}$

8 Model Evaluation

Estimator Evaluation Criteria

- Consistency: $\hat{\theta} \to \theta$ as $n \to \infty$
- Bias: $\mathbb{E}(\hat{\theta}) - \theta$
  - Unbiased: $\mathbb{E}(\hat{\theta}) = \theta$
  - Asymptotically unbiased: $\mathbb{E}[(\hat{\theta} - \theta)^2] = 0$ as $n \to \infty$
  - Asymptotically efficient: $\mathbb{E}[(\hat{\theta} - \theta)^2] = I$ as $n \to \infty$ where I is Fisher information (cf. Rao-Cramer bound)

Bias Variance Tradeoff

- Mean squared error $\mathbb{E}[(\hat{f}(X) - y)^2]$ can be decomposed into: $(\mathbb{E}[\hat{f}(X)] - f(X))^2 + \mathbb{V}(\hat{f}(X)) + \mathbb{E}[\epsilon^2] = \text{bias}^2 + \text{variance} + \text{irreducible error}$

  Proof:
  - $y = f(X) + \epsilon$
  - $\mathbb{E}[(\hat{f}(X) - y)^2] = \mathbb{E}[(\hat{f}(X) - \mathbb{E}[\hat{f}(X)] + \mathbb{E}[\hat{f}(X)] - f(X) + \epsilon)^2] = \mathbb{E}[(\hat{f}(X) - \mathbb{E}[\hat{f}(X)])^2] + \mathbb{E}[(\mathbb{E}[\hat{f}(X)] - f(X))^2] + \mathbb{E}[\epsilon^2] - 2\mathbb{E}[(\hat{f}(X) - \mathbb{E}[\hat{f}(X)])(\mathbb{E}[\hat{f}(X)] - f(X) + \epsilon)]$
  - Fourth term equals 0:
    - $2\mathbb{E}[(\hat{f}(X) - \mathbb{E}[\hat{f}(X)])(\mathbb{E}[\hat{f}(X)] - f(X) + \epsilon)] = 2(\mathbb{E}[\hat{f}(X)] - f(X) + \epsilon)\mathbb{E}[(\hat{f}(X) - \mathbb{E}[\hat{f}(X)])]$ because $(\mathbb{E}[\hat{f}(X)] - f(X) + \epsilon)$ is deterministic
    - In last equation, second term equals 0, so whole equation is 0
  - Then, we are left with: $\text{variance} + \text{bias}^2 + \text{irreducible error}$
- Bias: Error generated by the fact that we approximate a complex relationship via a simpler model (small function class) with a certain presupposed parametric form
- Variance: Error generated by the fact that we estimate the model parameters with a noisy training sample (small sample), rather than the population
- Irreducible error: Error generated by measurement error and the fact that we estimate y as a function of X, when it is a function of many other factors
- Bias variance tradeoff: Bias and variance cannot be reduced simultaneously
  - High variance associated with overfitting: Model corresponds too closely to particular training set resp. performs poorly on unseen data, but well on training set
  - High bias associated with underfitting: Model fails to capture underlying relationships resp. performs poorly on both training set and unseen data

Approximating Generalisation Loss via Empirical Loss

Via resampling methods —

Cross-validation:

- Partition data $\mathcal{Z}$ into K equally sized disjoint subsets: $\mathcal{Z} = \mathcal{Z}_1 \cup \mathcal{Z}_2 \cup ... \cup \mathcal{Z}_K$
- Produce estimator $\hat{f}^{-v}$ from $\mathcal{Z} \backslash \mathcal{Z}_v$ for $v \leq K$
- Empirical loss given by: $\hat{\mathcal{R}}^{cv} = \frac{1}{n}\sum_{i \leq n} LO(y_i - \hat{f}^{-k(i)}(x_i))$ where $k(i)$ maps i to partition $\mathcal{Z}_{k(i)}$ where (x_i, y_i) belongs

Bootstrapping:

- Draw B samples with replacement of size n from data $\mathcal{Z}$: $\mathcal{Z}^{*b}$
- Compute estimate $S(\mathcal{Z}^{*b})$ for each bootstrap sample
- For each estimate $S(\mathcal{Z}^{*b})$, we can give a mean and variance:
  - $\overline{S} = \frac{1}{B}\sum_b S(\mathcal{Z}^{*b})$
  - $\sigma^2(S) = \frac{1}{B-1}\sum_b (S(\mathcal{Z}^{*b}) - \overline{S})^2$
- Empirical loss given by: $\hat{\mathcal{R}}(\mathcal{A}) = \frac{1}{B}\sum_{b=1}^B \frac{1}{n}\sum_{i=1}^n LO(y_i - \hat{f}^{*b}(x_i))$
- Out-of-bag loss given by: $\hat{\mathcal{R}}^{bs} = \frac{1}{n}\sum_{i=1}^n \frac{1}{|C^{-i}|}\sum_{b \in C^{-i}} LO(y_i - \hat{f}^{*b}(x_i))$ where C^{-i} contains all bootstrap indices b so that $\mathcal{Z}^{*b}$ does not contain (x_i, y_i)
- Empirical loss of bootstrap uses training data to estimate $\hat{\mathcal{R}}$, i.e. it is generally too optimistic. We can correct this by combining the empirical and out-of-bag loss:
  - Probability that (x_i, y_i) is not in sample $\mathcal{Z}^{*b}$ of size n is given by $(1 - \frac{1}{n})^n = \frac{1}{e}$ as $n \to \infty \approx \frac{1}{3}$
  - Probability that (x_i, y_i) is in sample $\mathcal{Z}^{*b}$ of size n is given by $1 - \frac{1}{e}$ as $n \to \infty \approx \frac{2}{3}$
  - We then define: $\hat{\mathcal{R}}^{(0.632)} = 0.368\hat{\mathcal{R}}(\mathcal{A}) + 0.632\hat{\mathcal{R}}^{bs}$

9 Estimating Common Distributions

Gaussian

Frequentism (MLE) —

- Likelihood (excl. constants): $L = (\frac{1}{\sigma})^n \prod_{i=2}^n exp(-\frac{1}{2\sigma^2}(x^{(i)} - \mu)^2)$
- Log-likelihood: $LL = -nlog(\sigma) - \sum_{i=1}^n (\frac{1}{2\sigma^2}(x^{(i)} - \mu)^2)$
- μ_{MLE} is sample mean: $\frac{1}{n}\sum_{i=1}^n x^{(i)}$:
  - Derivative of log-likelihood wrt μ:
$$\nabla_\mu LL = \nabla_\mu - \sum_{i=1}^n (\frac{{x^{(i)}}^2 - 2x^{(i)}\mu + \mu^2}{2\sigma^2} = \nabla_\mu - \sum_{i=1}^n (-\frac{x^{(i)}\mu}{\sigma^2} + \frac{\mu^2}{2\sigma^2}) = -\sum_{i=1}^n (-\frac{x^{(i)}}{\sigma^2} + \frac{2\mu}{2\sigma^2}) = \sum_{i=1}^n (\frac{x^{(i)} - \mu}{\sigma^2}) = \sum_{i=1}^n x^{(i)} - n\mu = 0$$
- ${\sigma^2}_{MLE}$ is sample variance: $\frac{1}{n}\sum_{i=1}^n (x^{(i)} - \mu)^2$:
  - Derivative of log-likelihood wrt σ: $\nabla_\sigma LL = -n\nabla_\sigma log(\sigma) - \nabla_\sigma(\sum_{i=1}^n (\frac{(x^{(i)} - \mu)^2}{2\sigma^2}) =$

$\frac{-n}{\sigma} - \nabla_\sigma(\sum_{i=1}^n \frac{1}{2}\sigma^{-2}(x^{(i)} - \mu)^2) =$
$\frac{-n}{\sigma} - (\sum_{i=1}^n -1\sigma^{-3}(x^{(i)} - \mu)^2) =$
$-n + \sum_{i=1}^n (\frac{(x^{(i)} - \mu)^2}{\sigma^2}) = 0$

Bayesianism —

- Assume Σ is known and $\mu \sim \mathcal{N}(\mu_0, \Sigma_0)$ is the outcome of a random variable
- $p(\mu|X, \mu_0, \Sigma_0) \propto p(X|\mu, \Sigma) p(\mu|\mu_0, \Sigma_0)$
- $p(X|\mu, \Sigma) = \frac{1}{2\pi^{mn/2}} \frac{1}{|\Sigma|^{n/2}} exp(\frac{1}{2} \sum_{i=1}^n (x^{(i)} - \mu)^\top \Sigma^{-1} (x^{(i)} - \mu))$
- $p(\mu|\mu_0, \Sigma_0) = \frac{1}{2\pi^{m/2}} \frac{1}{|\Sigma_0|^{n/2}} exp(\frac{1}{2} \sum_{i=1}^n (\mu - \mu_0)^\top {\Sigma_0}^{-1} (\mu - \mu_0))$
- $p(\mu|X, \mu_0, \Sigma_0) \propto exp(-\frac{1}{2}(\mu^\top {\Sigma_0}^{-1} \mu + n\mu^\top \Sigma^{-1} \mu - 2{\mu_0}^\top {\Sigma_0}^{-1} \mu - 2n\overline{x}^\top \Sigma^{-1} \mu))$ after combining exponents of the prior and likelihood, expanding, absorbing terms unrelated to μ into a constant, and replacing $\sum_{i=1}^n x^{(i)\top}$ by $n\overline{x}^\top$
- We now apply a symmetric matrix property $x^\top A x + 2x^\top b = (x + A^{-1}b)^\top A (x + A^{-1}b) - b^\top A^{-1} b$, with $\mu = x$, $-({\Sigma_0}^{-1} + n\Sigma^{-1})^{-1} = A^{-1}$ and $(\Sigma^{-1} n\overline{x} + {\Sigma_0}^{-1} \mu_0) = b$
- Through this, we get $p(\mu|X, \mu_0, \Sigma_0) \propto \exp(\frac{1}{2}(\mu({\Sigma_0}^{-1} + n\Sigma^{-1})^{-1}(\Sigma^{-1} n\overline{x} + {\Sigma_0}^{-1}\mu_0))^\top ({\Sigma_0}^{-1} + n\Sigma^{-1})(\mu - ({\Sigma_0}^{-1} + n\Sigma^{-1})^{-1}(\Sigma^{-1} n\overline{x} + {\Sigma_0}^{-1}\mu_0))) = exp(\frac{1}{2}(\mu - \mu_n)^\top \Sigma_n^{-1} (\mu - \mu_n))$
- Thus, $p(\mu|X, \mu_0, \Sigma_0) \sim \mathcal{N}(\mu_n, \Sigma_n)$ with
  - $\mu_n = ({\Sigma_0}^{-1} + n\Sigma^{-1})^{-1}(\Sigma^{-1} n\overline{x} + {\Sigma_0}^{-1}\mu_0) =$ (if Σ equals 1) $\frac{n\overline{x}\Sigma_0 + \mu_0}{n\Sigma_0 + 1}$
  - $\Sigma_n = ({\Sigma_0}^{-1} + n\Sigma^{-1})^{-1} =$ (if Σ equals 1) $\frac{\Sigma_0}{n\Sigma_0 + 1}$
- For Bayesian parameter μ_n:
  - μ_n is a compromise between MLE and prior, approximating prior for small n and MLE for large n
  - If prior variance is small (i.e. if we are certain of our prior), prior mean weighs more strongly
- For Bayesian parameter Σ_n:
  - Σ_n approximates prior for small n and MLE for large n
  - If prior variance is small (i.e. if we are certain of our prior), posterior variance is also small

10 Linear Regression

Description

Task — Regression

Description —

- Supervised
- Parametric

Formulation

- $y^{(i)} = \beta \cdot x^{(i)}$ resp. $y = X\beta$ where X contains n rows, each of which represents an instance, and m columns, each of which represents a feature
- $\hat{y} = X\beta$ is a projection of y to the columnspace of X
- β lies in the rowspace of X resp. columnspace of $X^\top$

Optimization

Parameters — Find parameters β

Objective function — Ordinary least squares estimator (OLSE):

- Minimize mean squared error: $LO = \frac{1}{n}\sum_{i=1}^n (y^{(i)} - \beta \cdot x^{(i)})^2$ resp. $LO = (y - X\beta)^\top (y - X\beta)$

Orthogonality principle:

- Yields same result as OLSE
- $\hat{y} = X\beta$ is a projection of y to the columnspace of X
- Then, by the orthogonality principle, $X \cdot (\hat{y} - y) = X \cdot (X\beta - y) = 0$
- $\Rightarrow \beta = (X^\top X)^{-1} X^\top y$
- Alternatively, β lies in the columnspace of $X^\top$
- Then, we can express β as $X^\top [\alpha_1, ..., \alpha_n]^\top$
- This yields an equation system $y = XX^\top [\alpha_1, ..., \alpha_n]^\top$ which can be solved for α_i
- On that basis, β can be calculated

PCA:

- Yields same result as OLSE
- Instances $y^{(i)}, x^{(i)} = \xi^{(i)}$ can be projected onto hyperplane given by $X\beta$
- Projections are given by $\hat{\xi}^{(i)}$
- Residuals are given by $e^{(i)} = \xi^{(i)} - \hat{\xi}^{(i)}$
- Since $e^{(i)}$ is orthogonal to $\hat{\xi}^{(i)}$, we can write using Pythagorean theorem: $||e^{(i)}||^2 = ||\xi^{(i)}||^2 - ||\hat{\xi}^{(i)}||^2$
- This is a PCA via SVD problem

MLE:

- Yields same result as OLSE

Gradient descent:

- Minimum-norm solution
- Yields same result as OLSE

Pseudo Inverse:

- Yields same result as OLSE
- Minimum-norm solution
- β minimizes MSE if $\hat{y} = X\beta$ is a projection of y to the columnspace of X
- Given matrix projection via SVD, $XX^{\#}y$ is that projection
- $\Rightarrow \beta = X^{\#}y = (X^\top X)^{-1} X^\top y$

Optimization —

- $\nabla_\beta LO = \frac{1}{2}\nabla_\beta((y - \beta \cdot x)^2 = (y - \beta \cdot x)x = 0$ resp. $\nabla_\beta LO = \frac{1}{2}\nabla_\beta((y - X\beta)^\top (y - X\beta)) = \frac{1}{2}\nabla_\beta(\beta^\top X^\top X\beta - 2y^\top X\beta) = X^\top X\beta - X^\top y = X^\top (X\beta - y) = 0$
- $\Rightarrow \beta = (X^\top X)^{-1} X^\top y$

Hypothesis Testing of Found Parameters —

- Let $y|X \sim \mathcal{N}(y, \sigma^2 I) = \mathcal{N}(X\beta, \sigma^2 I)$
- Let $\hat{\beta} = (X^\top X)^{-1} X^\top y = X^+ y$ be the OLSE where X^+ is a scalar
- Then, $\hat{\beta} \sim \mathcal{N}(X^+ X\beta, X^{+\top}\sigma^2 X^+) = \mathcal{N}(\beta, (X^\top X)^{-1}\sigma^2)$
  Proof:
  - $\mathcal{N}(X^+ X\beta, X^{+\top}\sigma^2 X^+) = \mathcal{N}(I\beta, \sigma^2 X^+ X^{+\top})$ since X^+ is a scalar
  - Further, we have $\mathcal{N}(I\beta, \sigma^2 X^+((X^\top X)^{-1} X^\top)^\top) = \mathcal{N}(\beta, \sigma^2 X^+ X (X^\top X)^{-1\top}) = \mathcal{N}(\beta, \sigma^2 (X^\top X)^{-1})$ since $(X^\top X)$ is symmetric
- We can estimate σ^2 unbiasedly as: $\hat{\sigma}^2 = \frac{1}{n-m}\sum_{i \leq n}(X\hat{\beta} - y)^2$
- Then, confidence interval for $\hat{\beta}_j$ given by: $\hat{\beta}_j \pm z_{\alpha/2}\hat{se}(\hat{\beta}_j)$ where
  - $z_{\alpha/2} = \Phi^{-1}(\alpha/2)$ is Gaussian CDF
  - $\hat{se}(\hat{\beta}_j)$ is the j^{th} diagonal element of the covariance matrix $\sigma^2 (X^\top X)^{-1}$
- We can perform a hypothesis test on $\hat{\beta}$ with the *Wald test*:
  - $H_0 : \beta = \beta_0$ (typically 0)
    $H_1 : \beta \neq \beta_0$
  - Wald statistic: $W = \frac{\hat{\beta} - \beta_0}{\hat{se}}$
  - If p-value associated with W is smaller than α resp. if $|W|$ is greater than or equal to the critical value $z_{\alpha/2}$, we reject H_0

Evaluation —

- OLSE is unbiased if noise ϵ has zero mean:
  - Given $y = X\beta + \epsilon$, we can substitute $\hat{\beta} = (X^\top X)^{-1} X^\top (X\beta + \epsilon) = \beta + (X^\top X)^{-1} X^\top \epsilon$
  - Taking the expected value on both sides, we have: $\mathbb{E}(\hat{\beta}) = \beta + (X^\top X)^{-1} X^\top \mathbb{E}(\epsilon)$
  - Then, $\mathbb{E}(\hat{\beta}) = \beta$ if the noise has zero mean
- *Gauss Markov theorem*: OLSE is best (lowest variance, lowest MSE) unbiased estimator, if assumptions (X is full rank and there is no multicollinearity, heteroskedasticity, and exogeneity) are met
  Proof:
  - Let $A^\top y = (X^\top X)^{-1} X^\top y$ be the OLSE
  - Let $C^\top y$ be another unbiased estimator
  - $\mathbb{V}(A^\top y) = A^\top \mathbb{V}(y) A$ since A is constant
  - We can further develop to: $A^\top \sigma^2 I_m A = \sigma^2 A^\top A$ since variance is given by error term
  - Similarly, $\mathbb{V}(C^\top y) = \sigma^2 C^\top C$
  - For the OLSE, we can plug in $(X^\top X)^{-1} X^\top$ for A which yields: $\mathbb{V}(A^\top y) = \sigma^2 (X^\top X)^{-1} X^\top X (X^\top X)^{-1} = \sigma^2 (X^\top X)^{-1}$
  - Then, we have shown that $\mathbb{V}(A^\top y) \leq \mathbb{V}(C^\top y)$
- Nonetheless, there may be biased estimators that generate a lower variance and MSE

Characteristics —

- Convex with psd Hessian
- Has global minimum
- Has unique or infinitely many solutions
- Can be solved analytically, if $(X^\top X)$ is invertible
- If it has infinitely many solutions, the preferred solution is the *minimum-norm solution*, which minimizes $||\beta||$

11 Linear Minimum Mean Squared Error Estimation (LMMSE)

Description

- Minimizes mean squared error of two random variables, leveraging information about their mean and covariance
- Linear regression with large samples approximates LMMSE

Formulation

- y is observed
- x is a row vector and quantity of interest
- We estimate x as $\hat{x} = h^\top Y = \sum_i h_i y_i$ where X contains m rows, each of which represents the n-sized vector for a random variable
- This can be considered as a projection of x to the rowspace of Y

Optimization

Parameters — Find parameters h

Objective function —

- Minimize expected squared error: $LO = \mathbb{E}[|\hat{x} - x|]$

Optimization —

- By the orthogonality principle, $\mathbb{E}[(\hat{x} - x) \cdot y_i] = \mathbb{E}[(\sum_{l=1}^n h_l y_l - x) \cdot y_i] = 0$ for $i = 1, ..., n$
- Then, $\sum_{l=1}^n \mathbb{E}[y_l \cdot y_i] h_l = \mathbb{E}[x \cdot y_i]$ for $i = 1, ..., n$ which in matrix notation corresponds to $\begin{bmatrix} \mathbb{E}[y_1 \cdot y_1] & ... & \mathbb{E}[y_1 \cdot y_n] \\ ... & ... & ... \\ \mathbb{E}[y_n \cdot y_1] & ... & \mathbb{E}[y_n \cdot y_n] \end{bmatrix} \cdot \begin{bmatrix} h_1 \\ ... \\ h_n \end{bmatrix} = \begin{bmatrix} \mathbb{E}[x \cdot y_1] \\ ... \\ \mathbb{E}[x \cdot y_n] \end{bmatrix}$ resp. concisely $h^\top \mathbb{E}[YY^\top] = \mathbb{E}[xY^\top]$

12 Bayesian Linear Regression

Description

Task — Regression

Description —

- Supervised
- Parametric

Formulation

- $y^{(i)} = \beta \cdot x^{(i)}$ resp. $y = X\beta$
- $\beta \sim \mathcal{N}(0, T^2 I_m)$
- $p(\beta) \propto -\frac{1}{2T^2}\beta^\top \beta$

Optimization

Parameters — Find distribution of parameters β

Optimization —

- Likelihood:
  - Conditional on β, $y \propto \mathcal{N}(X\beta, \sigma^2 I_m)$
  - $p(y|X, \beta) = \frac{1}{(2\pi\sigma^2)^{n/2}} exp(-\frac{1}{2\sigma^2}(y - X\beta)^\top (y - X\beta))$
- Posterior: $p(\beta|X, y) \propto p(X, \beta) \times p(\beta) \propto exp(-\frac{1}{2\sigma^2}(y - X\beta)^\top (y - X\beta)) \times exp(-\frac{1}{2T^2}\beta^\top \beta) = exp(-\frac{1}{2}(\frac{1}{\sigma^2}y^\top y - 2\beta^\top X^\top y + \beta^\top X^\top X\beta) + \frac{1}{2T^2}\beta^\top \beta) \propto exp(-\frac{1}{2}(\beta^\top(\frac{1}{\sigma^2}X^\top X + \frac{1}{2T^2}I_m)\beta - \frac{2}{\sigma^2}\beta^\top X^\top y)$
- We now apply a symmetric matrix property $x^\top A x + 2x^\top b = (x + A^{-1}b)^\top A (x + A^{-1}b) - b^\top A^{-1} b$, with $\beta = x$, $(\frac{1}{\sigma^2}X^\top X + \frac{1}{2T^2}I_m) = A$ and $(\frac{1}{\sigma^2}X^\top y) = b$
- Through this, we get $p(\beta|X, y) \propto \exp(\frac{1}{2}(\beta + (\frac{1}{\sigma^2}X^\top X + \frac{1}{T^2}I_m)^{-1}(\frac{1}{\sigma^2}X^\top y))^\top (\frac{1}{\sigma^2}X^\top X + \frac{1}{T^2}I_m)(\beta + (\frac{1}{\sigma^2}X^\top X + \frac{1}{T^2}I_m)^{-1}(\frac{1}{\sigma^2}X^\top y)))$
- Thus, $p(\beta|X, y) \sim \mathcal{N}(\mu, \Sigma)$ with
  - $\mu = \Sigma \times \frac{1}{\sigma^2}X^\top y$
  - $\Sigma = (\frac{1}{\sigma^2}X^\top X + \frac{1}{T^2}I_m)^{-1}$
- Posterior mean corresponds to parameter β found by ridge regression, if $\lambda = \frac{\sigma^2}{T^2}$
- If we set an infinitely broad prior T^2 then the Bayesian estimate converges to the MLE estimate – if we have n = 0 training instances, the Bayesian estimate reverts to the prior

Characteristics —

- Convex with psd Hessian
- Has global minimum
- Can be solved analytically

13 Ridge (ℓ_2) Regression

Description

Task — Regression

Description —

- Supervised
- Parametric

Formulation

- $y^{(i)} = \beta \cdot x^{(i)}$ resp. $y = X\beta$

Optimization

Parameters — Find parameters β subject to $||\beta||^2 \leq t$ resp. $||\beta||^2 - t \leq 0$

Objective function —

- Minimize mean squared error subject to constraint
- Lagrangian formulation: $LO = \frac{1}{n}\sum_{i=1}^n (y^{(i)} - \beta \cdot x^{(i)})^2 + \lambda(||\beta||^2 - t)$ resp. $LO = (y - X\beta)^\top (y - X\beta) + \lambda(||\beta||^2 - t)$
- Still a OLSE problem, since we can rewrite the objective to minimize $(X\beta - y)$ as the objective to minimize $||(X'\beta - y')||^2$ with $X' = \begin{bmatrix} X \\ \lambda I \end{bmatrix}$ and $y' = \begin{bmatrix} y \\ 0 \end{bmatrix}$
- Corresponds to MAP estimation, when X is modeled as a vector of independent zero-mean Gaussian random variables

Optimization —

- $\nabla_\beta LO = 0$
- $\Rightarrow \beta = (X^\top X + \lambda I)^{-1} X^\top y$

Effect —

- Shrinks certain elements of β to near 0
  Proof:
  - Gradient at optimality given by $\frac{\partial(y - X\beta)^\top (y - X\beta)}{\partial\beta} + 2\lambda\beta = 0$
  - Then, $\beta^* = -\frac{1}{2\lambda}\frac{\partial(y - X\beta)^\top (y - X\beta)}{\partial\beta}$
  - This means that each parameter is shrunk by a factor determined by size of λ - the larger λ, the more the parameters are shrunk
  - Larger parameters experience a larger shrinkage

Characteristics —

- Strictly with pd Hessian, since Lagrangian term is strictly convex and the sum of a strictly convex function with a convex function is strictly convex

- Has global minimum
- Has unique solution, as $(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})$ has linearly independent columns
- Can be solved analytically, as $(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})$ is always invertible

14 Lasso (ℓ_1) Regression

Description

Task — Regression

Description —

- Supervised
- Parametric

Formulation

- $y^{(i)} = \boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)}$ resp. $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}$

Optimization

Parameters — Find parameters β subject to $|\beta| \leq t$ resp. $|\beta| - t \leq 0$

Objective function —

- Minimize mean squared error subject to constraint
- Lagrangian formulation: $LO = \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)})^2 + \lambda(|\boldsymbol{\beta}| - t)$ resp. $LO = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda(|\boldsymbol{\beta}| - t)$
- Corresponds to MAP estimation, when $\boldsymbol{X}$ is modeled as a vector of independent zero-mean Laplacian random variables

Effect —

- Shrinks certain elements of β to 0
  Proof:
  - Gradient at optimality given by $\frac{\partial(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})}{\partial\boldsymbol{\beta}} + \frac{\partial\lambda|\boldsymbol{\beta}|}{\partial\boldsymbol{\beta}} = 0$
  - $\frac{\partial\lambda|\boldsymbol{\beta}|}{\partial\boldsymbol{\beta}}$ non-differentiable because there is a sharp edge at $\beta = 0$, but we can work with subgradients for $\beta \neq 0$:

$$\frac{\partial}{\partial\beta}|\beta| = sgn(\beta) = \begin{cases} -1 & \beta < 0 \\ 0 & \beta = 0 \\ 1 & \beta > 0 \end{cases}$$

  - If we have $-\lambda < \frac{\partial(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})}{\partial\boldsymbol{\beta}} < \lambda$ the optimum is given by $\beta = 0$
  - This means that some parameters are set to 0 - the larger λ, the more parameters are set to 0
  - Small parameter values (i.e. unimportant features) are more likely to be set to 0
  - For parameters that are not set to 0, LASSO regression has a similar effect as ridge regression and shrinks these parameters towards 0

Characteristics —

- Convex, but not strictly convex
- Has global minimum
- Has unique or infinitely many solutions
- Cannot be solved analytically, since $|\boldsymbol{\beta}|$ is not differentiable at $\beta_i = 0$

15 Polynomial Regression

Description

Task — Regression

Description —

- Supervised
- Parametric

Formulation

- $y^{(i)} = \boldsymbol{\beta} \cdot \phi(\boldsymbol{x}^{(i)})$ resp. $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\beta}$ where $\boldsymbol{\Phi}$ is the transformed design matrix with rows $\phi(\boldsymbol{x}^{(i)})^\top$

Optimization

Parameters — Find parameters β

Objective function —

- Ordinary least squares estimator
- Minimize mean squared error

Optimization —

- $\nabla_{\boldsymbol{\beta}} LO = 0$
- $\Rightarrow \boldsymbol{\beta} = (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\boldsymbol{y}$

Characteristics —

- Convex with psd Hessian
- Has global minimum
- Has unique or infinitely many solutions
- Can be solved analytically, if $(\boldsymbol{\Phi}^\top\boldsymbol{\Phi})$ is invertible

16 Kernel Methods

Background on Kernel Methods

Description —

- Mechanism for tractably resp. implicitly mapping data into higher-dimensional feature space so that linear models can be used in this feature space
- To do so, we can employ the *kernel trick* and the *representer theorem*
- The requirements are that the kernel function fulfills *Mercer's theorem*, i.e. the kernel is a Mercer kernel

Kernel trick —

- Allows to operate in higher-dimensional feature space, without explicitly calculating this transformation, but instead implicitly computing the inner product in this feature space via a kernel function
- Given two inputs $\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}$ and a feature map $\varphi : \mathbb{R}^m \rightarrow \mathbb{R}^k$ we can define an inner product on $\mathbb{R}^k$ via the kernel function: $k(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = \varphi(\boldsymbol{x}^{(i)}) \cdot \varphi(\boldsymbol{x}^{(j)})$
- If a prediction function is described solely in terms of inner products in the input space, it can be lifted into the feature space by replacing the inner product with the kernel function
- Kernel trick cannot be used in conjunction with feature selection resp. sparsity inducing regularize (e.g. ℓ_1), as this does not satisfy the representer theorem

Representer theorem —

- Allows to avoid directly seeking the k parameters, but only the n parameters that characterize α
- Allows to avoid calculating $\varphi(\boldsymbol{z})$ when evaluating novel instance, but only sum over weighted set of n kernel function outputs

Mercer's theorem —

- Kernel function is psd and symmetric iff $k(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = \varphi(\boldsymbol{x}^{(i)}) \cdot \varphi(\boldsymbol{x}^{(j)})$
  - Psd: $\boldsymbol{x}^\top \boldsymbol{K} \boldsymbol{x} \geq 0$ where $\boldsymbol{K}$ is the kernel matrix
  - Symmetric: $k(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = k(\boldsymbol{x}^{(j)}, \boldsymbol{x}^{(i)})$
- Kernel that satisfies Mercer's theorem is a Mercer kernel, i.e. we can prove a kernel is a Mercer kernel either if it is psd and symmetric or by finding a feature map such that the kernel function corresponds to an inner product

Formulation

- Feature map $\varphi : \mathbb{R}^m \rightarrow \mathbb{R}^k$
- Linear prediction function: $\boldsymbol{\beta} \cdot \varphi(\boldsymbol{x}^{(i)})$
- Regularized loss function: $LO = \sum_{i=1}^{n} LO(y^{(i)}, \boldsymbol{\beta} \cdot \varphi(\boldsymbol{x}^{(i)}) + \Omega(\boldsymbol{\beta}))$
- Iff $\Omega(\boldsymbol{\beta}))$ is a non-decreasing function, then the parameters β that minimize the loss function can be rewritten as: $\boldsymbol{\beta} = \sum_{i=1}^{n} \alpha^{(i)}\varphi(\boldsymbol{x}^{(i)})$
- Outcome of novel instance can be predicted as: $\boldsymbol{\beta} \cdot \varphi(\boldsymbol{z}) = \sum_{i=1}^{n} \alpha^{(i)}\varphi(\boldsymbol{x}^{(i)}) \cdot \varphi(\boldsymbol{z}) = \sum_{i=1}^{n} \alpha^{(i)} k(\boldsymbol{x}^{(i)}, \boldsymbol{z})$
- Act of prediction becomes act of measuring similarity to training instances in feature map space

Kernel Types

Polynomial kernel —

- $\varphi(\boldsymbol{x}) = [x^\alpha]_{\alpha \in \mathbb{N}^m}$ where $\alpha = (\alpha_1, ..., \alpha_m)$ is the multi-index representing the power and $x^\alpha = x_1^{\alpha_1} \times ... \times x_m^{\alpha_m}$ is the mononomial term corresponding to the multi-index α
- E.g. if degree = 2, then $k(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = 1 + 2x_1^{(i)}x_1^{(j)} + 2x_2^{(i)}x_2^{(j)} + (x_1^{(i)}x_1^{(j)})^2 + (x_2^{(i)}x_2^{(j)})^2 + 2x_1^{(i)}x_1^{(j)}x_2^{(i)}x_2^{(j)}$
- Inner product diverges to infinity
- To address this, we often use RBF kernel instead

RBF kernel —

- Gives access to infinite feature space
- $\varphi(\boldsymbol{x}) = exp(-\frac{1}{2}||\boldsymbol{x}||^2)[\frac{\boldsymbol{x}^\alpha}{\sqrt{\alpha!}}]_{\alpha \in \mathbb{N}^m}$
- $k(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = \sigma^2 exp(-\frac{||\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}||^2}{2l^2})$
  Proof:
  - $exp(-\frac{1}{2}||\boldsymbol{x}^{(i)}||^2) exp(-\frac{1}{2}||\boldsymbol{x}^{(j)}||^2) \sum_\alpha [\frac{\boldsymbol{x}^{(i)\alpha}\boldsymbol{x}^{(j)\alpha}}{\alpha!}]$
  - Given multinomial series expansion, $\sum_\alpha [\frac{\boldsymbol{x}^{(i)\alpha}\boldsymbol{x}^{(j)\alpha}}{\alpha!}] = exp(\boldsymbol{x}^{(i)\top}\boldsymbol{x}^{(j)})$
  - $exp(-\frac{1}{2}||\boldsymbol{x}^{(i)}||^2 - \frac{1}{2}||\boldsymbol{x}^{(j)}||^2 + \boldsymbol{x}^{(i)\top}\boldsymbol{x}^{(j)}) = exp(-\frac{||\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}||^2}{2\sigma^2})$
- Length scale parameter l controls how quickly the similarity decays with distance: If l is large, points with high distance still have high covariance
- Variance parameter σ controls the vertical scale of the function
- RBF kernel is stationary, meaning that only the relative distance between two points determines the value output by the kernel function

Kernel compositions —

- New valid kernels can be composed via:
  - Addition: $k_1 + k_2$
  - Multiplication: $k_1 \times k_2$
  - Scaling: $c \times k_1$ for $c > 0$
  - Composition: $f(k_1)$ where f is a polynomial with positive coefficients or the exponential function
- Valid kernels:
  - $k'(x_1, x_2) = ck(x_1, x_2)$, since $\varphi'(x) = \sqrt{c}\varphi(x)$
  - $k'(x_1, x_2) = f(x_1)k(x_1, x_2)f(x_2)$, since $\varphi'(x) = f(x)\varphi(x)$
  - $k'(x_1, x_2) = k_1(x_1, x_2) + k_2(x_1, x_2)$, since the requirements for a valid kernel are that its psd and symmetric, which is retained when two psd and symmetric matrices are added
  - $k'(x_1, x_2) = k_1(x_1, x_2)k_2(x_1, x_2)$, since new kernel is given by the i^{th} feature value under feature map φ_1 multiplied by the j^{th} feature value under feature map φ_2
  - $k'(x_1, x_2) = exp(k(x_1, x_2))$, since we can apply Taylor series expansion $\sum_{n=1}^{r} \frac{k(x_1, x_2)^r}{r!} = exp(k(x_1, x_2)) = k'(x_1, x_2)$ as $r \rightarrow \infty$ and we know that exponentiation, addition, and scaling produces valid new kernels from above

17 Polynomial Kernel Regression

Description

Task — Regression

Description —

- Supervised
- Parametric

Formulation

- $\boldsymbol{y} = \boldsymbol{\beta} \cdot \varphi(\boldsymbol{x}^{(i)})$

Optimization

Parameters — Find parameters β

Objective function —

- Ordinary least squares estimator (OLSE)
- Minimize mean squared error: $LO = \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \boldsymbol{\beta} \cdot \varphi(\boldsymbol{x}^{(i)}))^2$

Optimization —

- Primal solution:
  - Parameters can be estimated as: $\boldsymbol{\beta} = (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\boldsymbol{y}$
  - Prediction for novel instance: $\boldsymbol{\beta} \cdot \varphi(\boldsymbol{z}) = (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\boldsymbol{y} \cdot \varphi(\boldsymbol{z}) = \boldsymbol{y}^\top\boldsymbol{\Phi}(\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\varphi(\boldsymbol{z})$
- Let us define $\boldsymbol{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^\top$ as the kernel matrix of the training data with $K_{ij} = \varphi(\boldsymbol{x}^{(i)}) \cdot \varphi(\boldsymbol{x}^{(j)})$
- Dual solution α if we have no regularization, i.e. $\lambda = 0$:
  - Parameters can be estimated as: $\boldsymbol{\beta} = \boldsymbol{\Phi}^\top\boldsymbol{K}^{-1}\boldsymbol{y}$
    Proof:
    - ⋆ $(\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \lambda\boldsymbol{I})\boldsymbol{\beta} = \boldsymbol{\Phi}^\top\boldsymbol{y}$
    - ⋆ $\Rightarrow \boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{\beta} + \lambda\boldsymbol{I}\boldsymbol{\beta} = \boldsymbol{\Phi}^\top\boldsymbol{y}$
    - ⋆ $\Rightarrow \boldsymbol{I}\boldsymbol{\beta} = \boldsymbol{\Phi}^\top\lambda^{-1}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\beta})$
    - ⋆ Since we know from the representer theorem that $\boldsymbol{\beta} = \boldsymbol{\Phi}^\top\boldsymbol{\alpha}$, we can say: $\boldsymbol{\alpha} = \lambda^{-1}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\beta})$
    - ⋆ We can further develop this to: $\lambda\boldsymbol{\alpha} = (\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\beta})$
    - ⋆ Replacing $\boldsymbol{\beta}$ by $\boldsymbol{\Phi}^\top\boldsymbol{\alpha}$ yields: $\lambda\boldsymbol{\alpha} = (\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Phi}^\top\boldsymbol{\alpha})$
    - ⋆ $\Rightarrow \boldsymbol{\alpha} = (\boldsymbol{\Phi}\boldsymbol{\Phi}^\top + \lambda\boldsymbol{I})^{-1}\boldsymbol{y} = \boldsymbol{K}^-1\boldsymbol{y}$
    - ⋆ With this, we can calculate the parameters: $\boldsymbol{\beta} = \boldsymbol{\Phi}^\top\boldsymbol{\alpha} = \boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top + \lambda\boldsymbol{I})^{-1}\boldsymbol{y} = \boldsymbol{\Phi}^\top\boldsymbol{K}^{-1}\boldsymbol{y}$
  - Prediction for novel instance: $\boldsymbol{\beta} \cdot \varphi(\boldsymbol{z}) = \boldsymbol{y}^\top(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top)^{-1}\boldsymbol{\Phi}\varphi(\boldsymbol{z}) = \boldsymbol{y}^\top(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top)^{-1}\boldsymbol{k}$ where $\boldsymbol{k} = \boldsymbol{\Phi}\varphi(\boldsymbol{z}) = [k(\boldsymbol{x}^{(1)}, \boldsymbol{z}), ..., k(\boldsymbol{x}^{(n)}, \boldsymbol{z})]^\top = [\varphi(\boldsymbol{x}^{(1)}) \cdot \varphi(\boldsymbol{z}), ..., \varphi(\boldsymbol{x}^{(n)}) \cdot \varphi(\boldsymbol{z})]^\top$ is a kernel vector, consisting of kernel values between training instances and new instance

Algorithm — Training:

1. Compute kernel matrix given RBF kernel
   Time complexity: $\mathcal{O}(n^2 \times m)$ for n^2 kernel matrix values and m number of features in each instance vector
2. Perform training by solving $\boldsymbol{\alpha} = \boldsymbol{K}^{-1}\boldsymbol{y}$ for $\boldsymbol{\alpha}$
   Time complexity: $\mathcal{O}(n^3)$
3. Store $\boldsymbol{\alpha}$
   Space complexity: $\mathcal{O}(n^2)$

Prediction:

1. Compute kernel vector
   Time complexity: $\mathcal{O}(n \times m \times d)$ for d new instances, given n instances in training data and m features in each instance vector
2. Store $\boldsymbol{k}$
   Space complexity: $\mathcal{O}(n \times d)$ for d new instances, given n as length of kernel vector
3. Predict response using stored kernel vector
   Time complexity: $\mathcal{O}(n \times d)$ for d new instances, given n as length of $\boldsymbol{\alpha}$

Value:

- Primal solution training is of time complexity $\mathcal{O}(k^3)$ and prediction is of time complexity $\mathcal{O}(k)$
- Dual solution speeds this up as seen above in the algorithm

Characteristics —

- Convex with psd Hessian
- Has global minimum
- Has unique or infinitely many solutions
- Can be solved analytically

18 Gaussian Processes

Description

Task — Models a distribution over functions

Description —

- Supervised
- Non-parametric

Formulation

- $y^{(i)} = \boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + \epsilon$ resp. $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \epsilon$
- $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{\Lambda}^{-1})$
- $\epsilon \sim \mathcal{N}(0, \sigma\boldsymbol{I}_m)$

Optimization

Optimization —

- If we compute the moment of the Gaussian:
  - $\mathbb{E}[\boldsymbol{y}] = \boldsymbol{X}^\top\mathbb{E}(\boldsymbol{\beta}) = \boldsymbol{X}^\top 0 = 0$
  - $\text{Cov}(\boldsymbol{y}) = \mathbb{E}[(\boldsymbol{X}^\top\boldsymbol{\beta} + \epsilon)(\boldsymbol{X}^\top\boldsymbol{\beta} + \epsilon)^\top] = \boldsymbol{X}\mathbb{E}(\boldsymbol{\beta}\boldsymbol{\beta}^\top) = \boldsymbol{X}^\top + \boldsymbol{X}\mathbb{E}(\boldsymbol{\beta})\mathbb{E}(\epsilon^\top) + \mathbb{E}(\epsilon)\mathbb{E}(\boldsymbol{\beta}^\top)\boldsymbol{X}^\top + \mathbb{E}(\epsilon\epsilon^\top) = 0$ where
    - ⋆ $\mathbb{V}(\boldsymbol{\beta}) = \mathbb{E}(\boldsymbol{\beta}\boldsymbol{\beta}^\top)$ and $\mathbb{V}(\epsilon) = \mathbb{E}(\epsilon\epsilon^\top)$ because $\mathbb{V}(x) = \mathbb{E}[(x - \mathbb{E}(x))^2] = \mathbb{E}[x^2]$ if $\mathbb{E}(x) = 0$, which is the case here due to the defined distributions
    - ⋆ $\mathbb{E}(\epsilon) = 0$
  - Plugging in the variance for β and ϵ, we have $\text{Cov}(\boldsymbol{y}) = \boldsymbol{X}\boldsymbol{\Lambda}^{-1}\boldsymbol{X}^\top + \sigma^2\boldsymbol{I}_m$

  - This can be written as a Kernel matrix $\boldsymbol{K}$:
    $\begin{bmatrix} K_{1,1}+\sigma^2 & ... & ... & K_{1,n} \\ ... & K_{2,2}+\sigma^2 & ... & ... \\ ... & ... & ... & ... \\ K_{n,1} & ... & ... & K_{n,n}+\sigma^2 \end{bmatrix}$
    with $K_{ij} = \boldsymbol{x}^{(\boldsymbol{i})\top}\boldsymbol{\Lambda}^{-1}\boldsymbol{x}^{(j)}$
  - In this kernel matrix, the kernel function can take any shape
- On this basis, Gaussian process is defined as collection of random variables such that every finite subset of variables is jointly Gaussian: $f \sim \mathcal{GP}(\boldsymbol{\mu}, \boldsymbol{K})$
- A new instance follows the distribution $p(y_{n+1}) = \mathcal{N}(\boldsymbol{k}^\top \boldsymbol{C}_n^{-1}\boldsymbol{y}, c - \boldsymbol{k}^\top \boldsymbol{C}_n^{-1}\boldsymbol{k})$ where
  - $\boldsymbol{k} = k(\boldsymbol{x}^{(\boldsymbol{1})}, \boldsymbol{x}^{(\boldsymbol{n+1})}), ..., k(\boldsymbol{x}^{(\boldsymbol{n})}, \boldsymbol{x}^{(\boldsymbol{n+1})})]^\top = [\varphi(\boldsymbol{x}^{(\boldsymbol{1})}) \cdot \varphi(\boldsymbol{x}^{(\boldsymbol{n+1})}), ..., \varphi(\boldsymbol{x}^{(\boldsymbol{n})}) \cdot \varphi(\boldsymbol{x}^{(\boldsymbol{n+1})})]^\top$ is the kernel vector
  - $\boldsymbol{C}_n = k(\boldsymbol{x}^{(\boldsymbol{i})}, \boldsymbol{x}^{(\boldsymbol{j})}) + \sigma^2 \boldsymbol{I}_m$
  - $c = k(\boldsymbol{x}^{(\boldsymbol{n+1})}, \boldsymbol{x}^{(\boldsymbol{n+1})}) + \sigma^2 \boldsymbol{I}_m$

  Proof:
  - We derive the joint distribution $p(\begin{bmatrix} \boldsymbol{y} \\ y_{n+1} \end{bmatrix}) \sim \mathcal{N}[0, \begin{bmatrix} \boldsymbol{C}_n | \boldsymbol{k} \\ \boldsymbol{k}^\top | c \end{bmatrix}]$
  - To obtain a closed-form solution for this, we can make use of the following theorem:
    - Given a joint Gaussian distribution: $\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}) \sim \mathcal{N}[\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} | \Sigma_{12} \\ \Sigma_{21} | \Sigma_{22} \end{bmatrix}]$
    - The conditional Gaussian distribution is given by: $p(a_2 | a_1 = z) = \mathcal{N}(u_2 + \Sigma_{21}\Sigma_{11}^{-1}(z - u_1), \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12})$
  - Then, we get $p(y_{n+1}) = \mathcal{N}(\boldsymbol{k}^\top \boldsymbol{C}_n^{-1}\boldsymbol{y}, c - \boldsymbol{k}^\top \boldsymbol{C}_n^{-1}\boldsymbol{k})$
- A new instance vector follows the distribution $p(\boldsymbol{y}_*) = \mathcal{N}(\boldsymbol{K}_{*n}\boldsymbol{K}_{nn}^{-1}\boldsymbol{y}_n, \boldsymbol{K}_{**} - \boldsymbol{K}_{*n}\boldsymbol{K}_{nn}^{-1}\boldsymbol{K}_{n*})$ where
  - Subscript $*$ indicates new instances, subscript n indicates old instances
  - $\boldsymbol{K}$is the kernel matrix, e.g. $\boldsymbol{K}_{*n}$ has new instances on rows and old instances on columns

Algorithm —

1. Compute kernel matrix based on observed data
2. Compute kernel vector based on observed data and new instance
3. Calculate mean and variance of predicted distribution
4. Return predicted distribution

19 SVM Classifier

General

Description

Task — Classification

Description —

- Supervised
- Parametric

Hard-Margin SVM Classifier

Formulation

- Assume $y \in \{-1, 1\}$
- Discriminant:
  - $f = sgn(\boldsymbol{\beta} \cdot \boldsymbol{x} + b)$
  - If sign is positive, f outputs 1, else -1
  - Separating hyperplane given by $\boldsymbol{z} = \boldsymbol{\beta} \cdot \boldsymbol{x} + b = 0$
  - Is a linear discriminant
  - $\boldsymbol{z} \perp \boldsymbol{\beta}$
- For some point $\tilde{x}$ closest to the origin:
  - The perpendicular distance to the origin is given by: $\boldsymbol{\beta} \cdot \tilde{\boldsymbol{x}} + b = 0$ since $\tilde{\boldsymbol{x}}$ lies on the separating hyperplane
  - Then, $||\boldsymbol{\beta}||\ ||\tilde{\boldsymbol{x}}|| cos(\varphi) + b = ||\boldsymbol{\beta}||\ ||\tilde{\boldsymbol{x}}||(-1) + b = 0$ because $\varphi = 180$ degrees
  - Then, $||\tilde{\boldsymbol{x}}|| = \frac{b}{||\boldsymbol{\beta}||}$
- For some point $\boldsymbol{x}^{(i)}$ above $\boldsymbol{z}$:
  - Projection of instance onto direction of $\boldsymbol{\beta}$: $\boldsymbol{x}^{(i)'} = \frac{\boldsymbol{x}^{(i)} \cdot \boldsymbol{\beta}}{||\boldsymbol{\beta}||^2}\boldsymbol{\beta}$
  - Distance of projection to the origin is given by $||\boldsymbol{x}^{(i)'}|| = cos(\varphi^{(i)})||\boldsymbol{x}^{(i)}|| = \frac{cos(\varphi^{(i)})||\boldsymbol{x}^{(i)}||\ ||\boldsymbol{\beta}||}{||\boldsymbol{\beta}||} = \frac{\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)}}{||\boldsymbol{\beta}||}$
  - Margin $\gamma^{(i)}$ of instance given by: $\gamma^{(i)} = ||\boldsymbol{x}^{(i)'}|| + ||\tilde{\boldsymbol{x}}|| = \frac{\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b}{||\boldsymbol{\beta}||}$
- For some point $\boldsymbol{x}^{(i)}$ below $\boldsymbol{z}$:
  - Margin $\gamma^{(i)}$ of instance given by: $\gamma^{(i)} = -\frac{\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b}{||\boldsymbol{\beta}||}$
- For well-classified points, $\gamma^{(i)} > 0$, for mis-classified points, $\gamma^{(i)} < 0$
- Given that $y \in \{-1, 1\}$ and thus $y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x} + b) > 0$: $\gamma^{(i)} = \frac{y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b)}{||\boldsymbol{\beta}||}$
- Margin of system defined by smallest margin for instance: $\gamma = min_i \gamma^{(i)} = \frac{1}{||\boldsymbol{\beta}||} min_i y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b)$
- Margin is invariate to scaling of $\boldsymbol{\beta}$ and b
- Thus, we can write:
  - $min_i \gamma^{(i)} = min_i y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b) = 1$
  - Then, $y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b) \geq 1$ (resp. $\geq m$ without scaling)
  - Moreover, since margin for system is defined by smallest margin for instance, $\gamma = \frac{1}{||\boldsymbol{\beta}||}$

Optimization

Parameters — Find parameters $\boldsymbol{\beta}$ and b

Objective function —

- Objective function: $\gamma = \frac{1}{||\boldsymbol{\beta}||}$ subject to $y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b) \geq 1$
- Equivalent cost function: $\gamma = \frac{1}{2}||\boldsymbol{\beta}||^2$ subject to $1 - y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b) \leq 0$
- Cost function in Lagrangian formulation: $\mathcal{L} = \frac{1}{2}||\boldsymbol{\beta}||^2 + \sum_{i=1}^n \alpha^{(i)}(1 - y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b))$
- By Slater's condition (linear separability), strong duality holds
- Objective function in dual Lagrangian: $\mathcal{D} = \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^n \alpha^{(i)}\alpha^{(j)}y^{(i)}y^{(j)}\boldsymbol{x}^{(i)}\boldsymbol{x}^{(j)} + \sum_{i=1}^n \sum_{j=1}^n \alpha^{(i)} - \alpha^{(i)}\alpha^{(j)}y^{(i)}y^{(j)}\boldsymbol{x}^{(i)}\boldsymbol{x}^{(j)} = \sum_{i=1}^n \alpha^{(i)} - \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^n \alpha^{(i)}\alpha^{(j)}y^{(i)}y^{(j)}\boldsymbol{x}^{(i)}\boldsymbol{x}^{(j)}$

Optimization —

- General solution:
  - $\nabla_{\boldsymbol{\beta}}\mathcal{L} = \boldsymbol{\beta} - \sum_{i=1}^n \alpha^{(i)}y^{(i)}\boldsymbol{x}^{(i)} = 0$
  - $\Rightarrow \boldsymbol{\beta}^* = \sum_{i=1}^n \alpha^{(i)}y^{(i)}$
  - $\nabla_b\mathcal{L} = -\sum_{i=1}^n \alpha^{(i)}y^{(i)} = 0$
  - Subject to:
    - $1 - y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b) \leq 0$
    - $\alpha^{(i)} \geq 0$
    - $\alpha^{(i)}(1 - y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b)) = 0$
- Dual optimization: Maximize α subject to
  - $\alpha^{(i)} \geq 0$
  - $\sum_{i=1}^n \alpha^{(i)}y^{(i)} = 0$ due to $\nabla_b\mathcal{L}$
- Note that only support vectors ($\alpha^{(i)} > 0$, sit on the hyperplane $1 = y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b) = 1$) matter in establishing $\boldsymbol{\beta}^*$:
  - Based on complementary slackness condition $\alpha^{(i)}(1 - y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b)) = 0$: We either have
    - $\alpha^{(i)} = 0$ and $(1 - y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b)) > 0$ resp.$y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b) > 1$ or
    - $\alpha^{(i)} > 0$ and $(1 - y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b)) = 0$ resp.$y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b) = 1$

Characteristics —

- Strictly convex with psd Hessian
- Has global minimum
- Has unique solution

Soft-Margin SVM Classifier

Optimization

Objective function —

- Cost function: Hinge loss: $max(0, 1 - \gamma^{(i)})$
- Mis-classified instances incur a loss
- Well-classified instances incur a loss, if their margin $\gamma^{(i)} < 1$
- Always is equal to or dominates the plain misclassification error
- To translate hinge loss into inequality constraint, we introduce slack variables $\xi^{(i)}$:
  - $y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b) \geq 1 - \xi^{(i)}$
  - By setting $\xi = 0$, we get pulled down towards hinge loss
  - For
    - Well-classified points outside of margin $\xi^{(i)} < 0$
    - Well-classified points within of margin $0 < \xi^{(i)} < 1$
    - Points on decision boundary $\xi^{(i)} = 1$
    - Mis-classified points $\xi^{(i)} > 1$
- We can then write cost function as slack variables penalized by ℓ_1 norm: $\frac{1}{2}||\boldsymbol{\beta}||^2 + C\sum_{i=1}^n \xi^{(i)}$ where
  - $||\boldsymbol{\beta}||^2$ maximizes margin and $C\sum_{i=1}^n \xi^{(i)}$ minimizes hinge loss
  - C is a hyperparameter that determines how tolerant we are of margin errors: If C is large, we are less tolerant, margin will decrease, and the soft-margin will become a hard-margin.

  subject to:
  - $1 - y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b) \leq \xi^{(i)}$ resp. $\xi^{(i)} + y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b) - 1 \geq 0$
  - $\xi^{(i)} \geq 0$
- Cost function in Lagrangian formulation: $\mathcal{L} = \frac{1}{2}||\boldsymbol{\beta}||^2 - \sum_{i=1}^n \alpha^{(i)}(y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b) - 1 + \xi^{(i)}) - \sum_{i=1}^n \zeta^{(i)}\xi^{(i)} + C\sum_{i=1}^n \xi^{(i)}$
- By Slater's condition (linear separability), strong duality holds
- Objective function in dual Lagrangian: $\mathcal{D} = \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^n \alpha^{(i)}\alpha^{(j)}y^{(i)}y^{(j)}\boldsymbol{x}^{(i)}\boldsymbol{x}^{(j)} + \sum_{i=1}^n \sum_{j=1}^n \alpha^{(i)} - \alpha^{(i)}\alpha^{(j)}y^{(i)}y^{(j)}\boldsymbol{x}^{(i)}\boldsymbol{x}^{(j)} = \sum_{i=1}^n \alpha^{(i)} - \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^n \alpha^{(i)}\alpha^{(j)}y^{(i)}y^{(j)}\boldsymbol{x}^{(i)}\boldsymbol{x}^{(j)}$

Optimization —

- General solution:
  - $\nabla_{\boldsymbol{\beta}}\mathcal{L} = \boldsymbol{\beta} - \sum_{i=1}^n \alpha^{(i)}y^{(i)}\boldsymbol{x}^{(i)} = 0$
  - $\Rightarrow \boldsymbol{\beta}^* = \sum_{i=1}^n \alpha^{(i)}y^{(i)}$
  - $\nabla_b\mathcal{L} = -\sum_{i=1}^n \alpha^{(i)}y^{(i)} = 0$
  - $\nabla_{\xi^{(i)}}\mathcal{L} = C - \alpha^{(i)} - \zeta^{(i)} = 0$
  - Subject to:
    - $-\xi^{(i)} - y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b) + 1 \leq 0$
    - $\xi^{(i)} \leq 0$
    - $\alpha^{(i)}, \zeta^{(i)} \geq 0$
    - $\alpha^{(i)}(-\xi^{(i)} - y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b) + 1) = 0$
    - $\zeta^{(i)}(-\xi^{(i)}) = 0$
- Dual optimization: Maximize α subject to
  - $\alpha^{(i)}, \zeta^{(i)} \geq 0$
  - $\sum_{i=1}^n \alpha^{(i)}y^{(i)} = 0$ due to $\nabla_b\mathcal{L}$
  - $0 \leq \alpha^{(i)} \leq C$ due to $\nabla_{\xi^{(i)}}\mathcal{L}$
- Note that only support vectors ($\alpha^{(i)} > 0$, sit in or on the hyperplane $1 \geq y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b) = 1$) matter in establishing $\boldsymbol{\beta}^*$:
  - Based on complementary slackness condition $\alpha^{(i)}(1 - y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b)) = 0$: We either have
    - $\alpha^{(i)} = 0$ and $(1 - y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b)) > 0$ resp.$y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b) > 1$ or
    - $\alpha^{(i)} > 0$ and $(1 - y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b)) = 0$ resp.$y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b) = 1$
  - Similarly, we either have
    - $\zeta^{(i)} = 0$ and $-\xi^{(i)} < 0$ resp.$\xi^{(i)} > 0$ or
    - $\zeta^{(i)} > 0$ and $-\xi^{(i)} = 0$ resp.$\xi^{(i)} = 0$
  - Then, each instance lies in one of three areas:
    - Beyond γ:
      - $\alpha^{(i)} = 0$
      - $C = \zeta^{(i)}$ due to $\nabla_{\xi^{(i)}}\mathcal{L}$
      - $\xi^{(i)} = 0$
      - $1 < y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b)$
    - On γ:
      - $\alpha^{(i)}, \zeta^{(i)} > 0$
      - $0 < \alpha^{(i)} < C$ due to $\nabla_{\xi^{(i)}}\mathcal{L}$
      - $\xi^{(i)} = 0$
      - $1 = y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b)$
    - Within γ:
      - $\alpha^{(i)} > 0$
      - $\zeta^{(i)} = 0$
      - $\alpha^{(i)} = C$ due to $\nabla_{\xi^{(i)}}\mathcal{L}$
      - $\xi^{(i)} > 0$
      - $1 > y^{(i)}(\boldsymbol{\beta} \cdot \boldsymbol{x}^{(i)} + b)$

Characteristics —

- Strictly convex with psd Hessian
- Has global minimum
- Has unique solution

20 PCA

Maximize Variance Approach

Description

Task — Dimensionality reduction via projection, create uncorrelated features

Description —

- Unsupervised
- Non-parametric

Overview — Identifies lower-dimensional subspace and projects data onto it such that the maximum amount of variance in the data is preserved. In lower-dimensional subspace:

- Axes are called *principal components*, where the first principal component is the axis accounting for the largest variance
- Each axis is given by an eigenvector with *loadings*, indicating how much each variable in the original data contributes to this eigenvector
- Variance captured along each axis is given by the corresponding eigenvalue

Formulation

- Project data $\{\boldsymbol{x}^{(\boldsymbol{i})}\}_{i=1}^n \in \mathbb{R}^m$ onto space $\mathbb{R}^d$ spanned by orthonormal basis $\{\boldsymbol{u}^{[j]}\}_{j=1}^d \in \mathbb{R}^m$ where $d << m$
- Each instance $\boldsymbol{x}^{(\boldsymbol{i})}$ is projected onto each basis vectors $\boldsymbol{u}^{[j]} \cdot \boldsymbol{x}^{(\boldsymbol{i})}$
- Each basis vector $\boldsymbol{u}^{[j]}$ contains m loadings $[u_i^{[j]}, ..., u_m^{[j]}]$, whose value indicates how important each feature m is for the j^{th} principal component
- Mean of projected data for a given basis vector: $\boldsymbol{u}^{[j]} \cdot \overline{\boldsymbol{X}} = \boldsymbol{u}^{[j]} \cdot \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}^{(\boldsymbol{i})}$
- Variance of projected data for a given basis vector: $\frac{1}{n}\sum_{i=1}^n (\boldsymbol{u}^{[j]} \cdot \boldsymbol{x}^{(\boldsymbol{i})} - \boldsymbol{u}^{[j]} \cdot \overline{\boldsymbol{X}})^2 = \boldsymbol{u}^{[j]\top}\boldsymbol{S}\boldsymbol{u}^{[j]}$ where $\boldsymbol{S}$ is the covariance matrix $\boldsymbol{S} = \frac{1}{n}\sum_{i=1}^n (\boldsymbol{x}^{(\boldsymbol{i})} - \overline{\boldsymbol{X}})(\boldsymbol{x}^{(\boldsymbol{i})} - \overline{\boldsymbol{X}})^\top = \frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}$

Optimization

Parameters — Find $\{\boldsymbol{u}^{[j]}\}_{j=1}^d$

Objective function —

- Maximize variance $\sum_{j=1}^d \boldsymbol{u}^{[j]\top}\boldsymbol{S}\boldsymbol{u}^{[j]}$ subject to orthonormal $\{\boldsymbol{u}^{[j]}\}_{j=1}^d$
- Gives rise to Lagrangian formulation
- Lagrangian formulation for $\boldsymbol{u}^{[1]}$ capturing the most variance: $\mathcal{L} = \boldsymbol{u}^{[\boldsymbol{1}]\top}\boldsymbol{S}\boldsymbol{u}^{[\boldsymbol{1}]} - \lambda^{[1]}(\boldsymbol{u}^{[\boldsymbol{1}]} \cdot \boldsymbol{u}^{[\boldsymbol{1}]} - 1)$ where $\lambda^{[1]}$ captures the orthonormality constraint that $\boldsymbol{u}^{[\boldsymbol{1}]} \cdot \boldsymbol{u}^{[\boldsymbol{1}]} = 1$

- Lagrangian formulation for $u^{[2]}$ capturing the secondmost variance: $\mathcal{L} = \boldsymbol{u}^{[2]\top} S \boldsymbol{u}^{[2]} - \lambda^{[2]}(\boldsymbol{u}^{[2]} \cdot \boldsymbol{u}^{[2]} - 1) - \lambda^{[1][2]}(\boldsymbol{u}^{[1]} \cdot \boldsymbol{u}^{[2]} - 0)$ where $\lambda^{[1][2]}$ captures the orthogonality constraint that $\boldsymbol{u}^{[1]} \cdot \boldsymbol{u}^{[2]} = 0$

Optimization — For $u^{[1]}$:

- $\nabla_{\boldsymbol{u}^{[1]}} \mathcal{L} = 2S\boldsymbol{u}^{[1]} - 2\lambda^{[1]}\boldsymbol{u}^{[1]} = 0$
- $\Rightarrow S\boldsymbol{u}^{[1]} = \lambda^{[1]}\boldsymbol{u}^{[1]}$
- This is the eigenvector/eigenvalue equation, so $u^{[1]}$ is the eigenvector of S and $\lambda^{[1]}$ is the associated eigenvalue
- We see that the variance of the projected data is equal to $\lambda^{[1]}$: $\boldsymbol{u}^{[1]\top} S \boldsymbol{u}^{[1]} = \boldsymbol{u}^{[1]\top} \lambda^{[1]} \boldsymbol{u}^{[1]} = \lambda^{[1]} \boldsymbol{u}^{[1]\top} \boldsymbol{u}^{[1]} = \lambda^{[1]} \times 1$

For $u^{[2]}$:

- $\nabla_{\boldsymbol{u}^{[2]}} \mathcal{L} = 2S\boldsymbol{u}^{[2]} - 2\lambda^{[2]}\boldsymbol{u}^{[2]} - \lambda^{[1][2]}\boldsymbol{u}^{[1]} = 0$
- $\Rightarrow S\boldsymbol{u}^{[2]} = \lambda^{[2]}\boldsymbol{u}^{[2]}$
  Proof:
  - Multiplying with $\boldsymbol{u}^{[1]\top}$: $2\boldsymbol{u}^{[1]\top} S \boldsymbol{u}^{[2]} - 2\lambda^{[2]}\boldsymbol{u}^{[1]\top}\boldsymbol{u}^{[2]} - \lambda^{[1][2]}\boldsymbol{u}^{[1]\top}\boldsymbol{u}^{[1]} = 0$
  - $= 2\boldsymbol{u}^{[1]\top} S \boldsymbol{u}^{[2]} - 0 - \lambda^{[1][2]} \times 1 = 0$ because of orthogonality resp. orthonormality
  - $= 2\boldsymbol{u}^{[2]\top} S \boldsymbol{u}^{[1]} - \lambda^{[1][2]} = 0$ because the variance is a scalar and can be transposed and because the covariance matrix is symmetric
  - $= 2\boldsymbol{u}^{[2]\top} \lambda^{[1]} \boldsymbol{u}^{[1]} - \lambda^{[1][2]} = 0$ after plugging in the first found basis vector
  - $= 2\lambda^{[1]} \times 0 - \lambda^{[1][2]} = 0$
  - $= \lambda^{[1][2]} = 0$

... continue as for previous vector

In the end, we have a total projected variance of $\sum_{j=1}^{d} \lambda^{[j]}$

Characteristics —

- Convex
- Has global minimum
- Has unique or infinitely many solutions
- Can be solved analytically

SVD Approach

Formulation

- Project data $\{\boldsymbol{x}^{(i)}\}_{i=1}^{n} \in \mathbb{R}^m$ onto space $\mathbb{R}^d$ spanned by orthonormal basis $\{\boldsymbol{u}^{[j]}\}_{j=1}^{d} \in \mathbb{R}^m$ where $d << m$
- $\tilde{\boldsymbol{x}}^{(i)} = \sum_{j=1}^{d} \alpha_{ij}\boldsymbol{u}^{[j]} + \sum_{j=d+1}^{m} \gamma_j \boldsymbol{u}^{[j]}$ where α_{ij} is particular to the instance, γ_j is the same for all instances and maps up to the subspace

Optimization

Objective function —

- Reconstruction error:
  - $J = \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}^{(i)} - \tilde{\boldsymbol{x}}^{(i)}\|^2 = \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}^{(i)}\|^2 - \|\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{x}^{(i)}\|^2 = \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}^{(i)}\|^2 - \|\boldsymbol{B}^T\boldsymbol{x}^{(i)}\|^2$ where the last step follows since $\boldsymbol{B}$ is orthogonal
  - $\boldsymbol{x}^{(i)}$ is a column of $X^\top$
  - If $A = X^\top$ and SVD of A is $\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$, then $\boldsymbol{B}$ is given by $\boldsymbol{U}^{(j \leq d)}$, i.e. the first d columns of $\boldsymbol{U}$
  - Then, reconstruction error is given by: $J = \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}^{(i)}\|^2 - \|\boldsymbol{B}^T\boldsymbol{x}^{(i)}\|^2 = \frac{1}{d}\sum_{i=1}^{d} \sigma_d^2$

21 GMM

Description

Task — Clustering

Description —

- Unsupervised
- Non-parametric

Formulation

- Instances $\{\boldsymbol{x}^{(i)}\}_{i=1}^{n}$
- Each instance has a latent cluster assignment given by: $z^{(i)} \in \{1, ..., k\}$
- Probability that cluster assigned to instance i is cluster j is given by: $\pi^{[j]} = p(z^{(i)} = j)$
- Contingent on cluster assignment, each instance is the outcome of a random variable associated with a given cluster: $\boldsymbol{x}^{(i)}|z^{(i)} = j \sim \mathcal{N}(\boldsymbol{x}^{(i)}|\boldsymbol{\mu}^{[j]}, \Sigma^{[j]})$ where $\boldsymbol{\mu}^{[j]}$ is the mean and $\Sigma^{[j]}$ is the covariance associated with cluster j
- Then, marginal distribution of each instance is given by: $p(\boldsymbol{x}^{(i)}) = \sum_{j=1}^{k} \mathcal{N}(\boldsymbol{x}^{(i)}|\boldsymbol{\mu}^{[j]}, \Sigma^{[j]}) \times \pi^{[j]}$
- This is the GMM, characterized by parameters $\{\boldsymbol{\mu}^{[j]}, \Sigma^{[j]}, \pi^{[j]}\}_{j=1}^{k}$

Optimization

Parameters — Find parameters $\{\boldsymbol{\mu}^{[j]}, \Sigma^{[j]}, \pi^{[j]}\}_{j=1}^{k}$

Objective function —

- Maximize likelihood $L = \sum_{i=1}^{n} log(\sum_{j=1}^{k} \mathcal{N}(\boldsymbol{x}^{(i)}|\boldsymbol{\mu}^{[j]}, \Sigma^{[j]}) \times \pi^{[j]})$ subject to $\sum_{j=1}^{k} \pi^{[j]} = 1$ and $\Sigma^{[j]} > 0$
- This is a constrained, not concave, not analytically solvable optimization problem
- Temporarily assume we know which cluster each instance is associated with
- Let us define a distribution q over $1, ..., k$: $q(z^{(i)}) = p(z^{(i)} = j|\boldsymbol{x}^{(i)}, \theta^{(t)}) = \frac{\mathcal{N}(\boldsymbol{x}^{(i)}|\boldsymbol{\mu}^{[j](t)}, \Sigma^{[j](t)}) \times \pi^{[j](t)}}{\sum_{j=1}^{k} \mathcal{N}(\boldsymbol{x}^{(i)}|\boldsymbol{\mu}^{[j](t)}, \Sigma^{[j](t)}) \times \pi^{[j](t)}}$
- Then, we can rewrite log likelihood as: $L = \sum_{i=1}^{n} log(\sum_{j=1}^{k} q(z^{(i)}) \frac{\mathcal{N}(\boldsymbol{x}^{(i)}|\boldsymbol{\mu}^{[j]}, \Sigma^{[j]}) \times \pi^{[j]}}{q(z^{(i)})})$
- According to Jensen's inequality: $L = \sum_{i=1}^{n} log(\sum_{j=1}^{k} q(z^{(i)}) \frac{\mathcal{N}(\boldsymbol{x}^{(i)}|\boldsymbol{\mu}^{[j]}, \Sigma^{[j]}) \times \pi^{[j]}}{q(z^{(i)})}) \geq \sum_{i=1}^{n} \sum_{j=1}^{k} q(z^{(i)}) log(\frac{\mathcal{N}(\boldsymbol{x}^{(i)}|\boldsymbol{\mu}^{[j]}, \Sigma^{[j]}) \times \pi^{[j]}}{q(z^{(i)})})$
- RHS can be rewritten: $\mathbb{E}_q[log(p_\theta(\boldsymbol{x}^{(i)}))] = \mathbb{E}_q[log(\frac{p_\theta(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)})}{p_\theta(\boldsymbol{z}^{(i)}|\boldsymbol{x}^{(i)})} \frac{q(z^{(i)})}{q(z^{(i)})})] = \mathbb{E}_q[log(\frac{p_\theta(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)})}{q(z^{(i)})})] + \mathbb{E}_q[log(\frac{q(z^{(i)})}{p_\theta(\boldsymbol{z}^{(i)}|\boldsymbol{x}^{(i)})})] = M + E$
- E corresponds to the KL divergence between $q(z^{(i)})$ and $p(z^{(i)} = j|\boldsymbol{x}^{(i)})$
- $L \geq M \Leftrightarrow E \geq 0$, which we can show to be the case:
  - $E = \mathbb{E}_q[log(\frac{q(z^{(i)})}{p_\theta(\boldsymbol{z}^{(i)}|\boldsymbol{x}^{(i)})})] = \mathbb{E}_q[-log(\frac{p_\theta(\boldsymbol{z}^{(i)}|\boldsymbol{x}^{(i)})}{q(z^{(i)})})]$
  - According to Jensen's inequality: $E \geq -log(\mathbb{E}_q[\frac{p_\theta(\boldsymbol{z}^{(i)}|\boldsymbol{x}^{(i)})}{q(z^{(i)})}]) = -log(\sum_{i=1}^{k} q(z^{(i)} \frac{p_\theta(\boldsymbol{z}^{(i)}|\boldsymbol{x}^{(i)})}{q(z^{(i)})}) = -log(\sum_{i=1}^{k} p_\theta(\boldsymbol{z}^{(i)}|\boldsymbol{x}^{(i)})) = -log(1) = 0$
- $L = M \Leftrightarrow E = 0$, i.e. when $q(z^{(i)}) = p(z^{(i)} = j|\boldsymbol{x}^{(i)}, \theta^{(t)})$
- Then, we have a lower bound on L, provided by M, with equality to M, if we set q correspondingly
- M is tractable to optimize, since the logarithm now only contains a product, not a sum, and can be decomposed: $log(p_\theta(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)})) = log(\mathcal{N}(\boldsymbol{x}^{(i)}|\boldsymbol{\mu}^{[j]}, \Sigma^{[j]}) \times \pi^{[j]}) = log(\mathcal{N}(\boldsymbol{x}^{(i)}|\boldsymbol{\mu}^{[j]}, \Sigma^{[j]})) + log(\pi^{[j]})$

Optimization — *Expectation maximization algorithm*

1. Randomly initialize $\theta^{(t)} = \{\boldsymbol{\mu}^{[j](t)}, \Sigma^{[j](t)}, \pi^{[j](t)}\}_{j=1}^{k}$
2. *E-step*: Minimize E, by computing $q(z^{(i)})$ given $\boldsymbol{x}^{(i)}$ and $\theta^{(t)}$
3. *M-step*: Maximize M, by updating $\theta^{(t)}$ based on MLE for Gaussians, while keeping $q(z^{(i)})$ fixed:
   - $\boldsymbol{\mu}^{[j](t+1)} = \frac{\sum_{i=1}^{n} q(z^{(i)})\boldsymbol{x}^{(i)}}{\sum_{i=1}^{n} q(z^{(i)})}$
   - $\Sigma^{[j](t+1)} = \frac{\sum_{i=1}^{n} q(z^{(i)})(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{[j](t+1)})(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{[j](t+1)})^\top}{\sum_{i=1}^{n} q(z^{(i)})}$
   - $\pi^{[j](t+1)} = \frac{1}{n}\sum_{i=1}^{n} q(z^{(i)})$
4. Repeat 2 and 3 until convergence

Characteristics —

- Not convex
- May converge to local minimum
- Not analytically solvable
- Always converges, since $L \geq M$ and $M^{(t+1)} \geq M^{(t)}$ due to maximizing over M at each step

22 Bayesian Neural Networks

Setting

- In Bayesian setting, normalization constant is computationally intractable

Formulation

Since original setting is computationally intractable, we can turn to *variational inference*:

- Variational inference approximates true posterior $p(w|\boldsymbol{D})$ by simpler, parametrized distribution $q(w|\boldsymbol{\theta})$
- We assume $q(w|\boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$ with $\boldsymbol{\theta} = (\boldsymbol{\mu}, \sigma)$

Optimization

Parameters — Find parameters θ

Objective function —

- Minimize KL divergence: $\boldsymbol{\theta}^* = argmin_{\boldsymbol{\theta}} KL[q(\boldsymbol{w}|\boldsymbol{\theta})\|p(\boldsymbol{w}|\boldsymbol{D})] = argmin_{\boldsymbol{\theta}} \mathbb{E}_{w\sim q} log(q(\boldsymbol{w}|\boldsymbol{\theta})) - \mathbb{E}_{w\sim q} log(p(\boldsymbol{D}|\boldsymbol{w})) - \mathbb{E}_{w\sim q} log(p(\boldsymbol{w}))$
  Proof:
  - $argmin_{\boldsymbol{\theta}} KL[q(\boldsymbol{w}|\boldsymbol{\theta})|p(\boldsymbol{w}|\boldsymbol{D})] = argmin_{\boldsymbol{\theta}} \mathbb{E}_{w\sim q}[log(\frac{q(\boldsymbol{w}|\boldsymbol{\theta})}{p(\boldsymbol{w}|\boldsymbol{D})})] = argmin_{\boldsymbol{\theta}} \mathbb{E}_{w\sim q}[log(q(\boldsymbol{w}|\boldsymbol{\theta}))] - \mathbb{E}_{w\sim q}[log(p(\boldsymbol{w}|\boldsymbol{D}))] = argmin_{\boldsymbol{\theta}} \mathbb{E}_{w\sim q}[log(q(\boldsymbol{w}|\boldsymbol{\theta}))] - \mathbb{E}_{w\sim q}[\frac{p(\boldsymbol{D}|\boldsymbol{w}) \times p(\boldsymbol{w})}{p(\boldsymbol{D})}] = argmin_{\boldsymbol{\theta}} \mathbb{E}_{w\sim q} log(q(\boldsymbol{w}|\boldsymbol{\theta})) - \mathbb{E}_{w\sim q} log(p(\boldsymbol{D}|\boldsymbol{w})) - \mathbb{E}_{w\sim q} log(p(\boldsymbol{w})) + \mathbb{E}_{w\sim q} log(p(\boldsymbol{D})) = argmin_{\boldsymbol{\theta}} \mathbb{E}_{w\sim q} log(q(\boldsymbol{w}|\boldsymbol{\theta})) - \mathbb{E}_{w\sim q} log(p(\boldsymbol{D}|\boldsymbol{w})) - \mathbb{E}_{w\sim q} log(p(\boldsymbol{w})) + const.$

Optimization —

- To calculate gradient, we can leverage the *reparametrization trick*
- $\frac{\partial}{\partial \boldsymbol{\theta}} \mathbb{E}_{w\sim q}[log(q(\boldsymbol{w}|\boldsymbol{\theta})) - log(p(\boldsymbol{D}|\boldsymbol{w})) - log(p(\boldsymbol{w}))] = \frac{\partial}{\partial \boldsymbol{\theta}} \mathbb{E}_{w\sim q}[F(\boldsymbol{w}, \boldsymbol{\theta})]$ can be reparametrized to:
  - $\frac{\partial}{\partial \boldsymbol{\mu}} \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\boldsymbol{I})}[\frac{\partial}{\partial \boldsymbol{w}} F(\boldsymbol{w}, \boldsymbol{\theta}) + \frac{\partial}{\partial \boldsymbol{\mu}} F(\boldsymbol{w}, \boldsymbol{\theta})]$
  - $\frac{\partial}{\partial \sigma} \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\boldsymbol{I})}[\epsilon^\top \frac{\partial}{\partial \boldsymbol{w}} F(\boldsymbol{w}, \boldsymbol{\theta}) + \frac{\partial}{\partial \sigma} F(\boldsymbol{w}, \boldsymbol{\theta})]$
- To optimize this, we can use gradient descent with the following algorithm:
  1. Initialize μ and σ
  2. For $t = 1, 2, ...$
     (a) Sample $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$
     (b) Compute $F(w, \theta)$
     (c) $\boldsymbol{\mu}_{t+1} \leftarrow \boldsymbol{\mu}_t - \eta_t[\frac{\partial}{\partial \boldsymbol{w}} F(\boldsymbol{w}, \boldsymbol{\theta}) + \frac{\partial}{\partial \boldsymbol{\mu}} F(\boldsymbol{w}, \boldsymbol{\theta})]|_{\boldsymbol{\mu}=\boldsymbol{\mu}_t}$
     (d) $\sigma_{t+1} \leftarrow \sigma_t - \eta_t[\epsilon^\top \frac{\partial}{\partial \boldsymbol{w}} F(\boldsymbol{w}, \boldsymbol{\theta}) + \frac{\partial}{\partial \sigma} F(\boldsymbol{w}, \boldsymbol{\theta})]|_{\sigma=\sigma_t}$

23 Other

ML Models

Score — The score is the derivative of the log-likelihood: $\Lambda = \frac{\partial}{\partial \boldsymbol{\theta}} log(p(\boldsymbol{x}|\boldsymbol{\theta})) = \frac{\frac{\partial}{\partial \boldsymbol{\theta}} p(\boldsymbol{x}|\boldsymbol{\theta})}{p(\boldsymbol{x}|\boldsymbol{\theta})}$

The expected score is given by:

$\mathbb{E}(\Lambda) = \int p(\boldsymbol{x}|\boldsymbol{\theta}) \frac{\frac{\partial}{\partial \boldsymbol{\theta}} p(\boldsymbol{x}|\boldsymbol{\theta})}{p(\boldsymbol{x}|\boldsymbol{\theta})} dx = \frac{\partial}{\partial \boldsymbol{\theta}} \int p(\boldsymbol{x}|\boldsymbol{\theta}) dx = \frac{\partial}{\partial \boldsymbol{\theta}} \times 1 = 0$

Fisher information —

- $\boldsymbol{I} = \mathbb{E}[(\Lambda)^2] = \mathbb{E}[(\frac{\partial}{\partial \boldsymbol{\theta}} log(p(\boldsymbol{x}|\boldsymbol{\theta})))^2] = \mathbb{V}(\frac{\partial log(p(\boldsymbol{x}|\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}})$ where Λ is the score
- Equality is given because $\mathbb{V}(x) = \mathbb{E}[(x - \mathbb{E}(x))^2] = \mathbb{E}[x^2]$ if $\mathbb{E}(x) = 0$, which is the case here, given that the expected score is 0

Rao-Cramer bound —

- Shows that there does not exist an asymptotically unbiased parameter estimator
- For each unbiased estimator, $\mathbb{E}[(\hat{\theta} - \theta)^2] \geq \frac{1}{I}$ where I is the Fisher information
- For estimators in general, $\frac{(\frac{\partial}{\partial \boldsymbol{\theta}} \text{bias} + 1)^2}{\boldsymbol{I}} + \text{bias}^2 \leq \mathbb{E}[(\hat{\theta} - \theta)^2]$, so there is a trade-off if the bias derivative is negative and the squared bias is positive, whereby a biased estimator may produce better results than an unbiased estimator

Proof:

- Given Cauchy Schwarz inequality, we can say: $\mathbb{E}[(\Lambda - \mathbb{E}((\Lambda))(\hat{\theta} - \mathbb{E}(\hat{\theta}))]^2 \leq \mathbb{E}[(\Lambda - \mathbb{E}((\Lambda))^2]\mathbb{E}[(\hat{\theta} - \mathbb{E}(\hat{\theta}))^2]$ where Λ is the score
- We know that $\mathbb{E}(\Lambda) = 0$
- Let's look at the LHS of the equation:
  - Since $\mathbb{E}(\Lambda) = 0$, we can simplify to $\mathbb{E}[\Lambda(\hat{\theta} - \mathbb{E}(\hat{\theta}))] = \mathbb{E}[\Lambda\hat{\theta}] - \mathbb{E}[\Lambda]\mathbb{E}[\hat{\theta}] = \mathbb{E}[\Lambda\hat{\theta}] - 0$
  - This can be developed to: $\mathbb{E}[\Lambda\hat{\theta}] = \int p(\boldsymbol{x}|\boldsymbol{\theta}) \frac{\frac{\partial}{\partial \boldsymbol{\theta}} p(\boldsymbol{x}|\boldsymbol{\theta})}{p(\boldsymbol{x}|\boldsymbol{\theta})} \hat{\theta} dx = \frac{\partial}{\partial \boldsymbol{\theta}} (\int p(\boldsymbol{x}|\boldsymbol{\theta}) \hat{\theta} dx - \theta) + 1$ where the last part $(-\theta) + 1$) can be added, because $\frac{\partial}{\partial \boldsymbol{\theta}} - \theta = -1$ and we compensate this with $+1$
  - This is equal to the derivative of the bias + 1: $\frac{\partial}{\partial \boldsymbol{\theta}} (\int p(\boldsymbol{x}|\boldsymbol{\theta}) \hat{\theta} dx - \theta) + 1 = \frac{\partial}{\partial \boldsymbol{\theta}} (\mathbb{E}[\hat{\theta}] - \theta) + 1 = \frac{\partial}{\partial \boldsymbol{\theta}} \text{bias} + 1$
- Let's look at the RHS of the equation: Since $\mathbb{E}(\Lambda) = 0$, first term is $\mathbb{E}(\Lambda^2) = \boldsymbol{I}$
- Then, we have: $(\frac{\partial}{\partial \boldsymbol{\theta}} \text{bias} + 1)^2 \leq \boldsymbol{I} \times \mathbb{E}[(\hat{\boldsymbol{\theta}} - \mathbb{E}(\hat{\boldsymbol{\theta}}))^2] = \boldsymbol{I} \times \mathbb{E}[(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta} - \mathbb{E}(\hat{\boldsymbol{\theta}}) + \boldsymbol{\theta})^2] = ... = \boldsymbol{I} \times \mathbb{E}[(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^2] - \text{bias}^2$
- Then, we have $\frac{(\frac{\partial}{\partial \boldsymbol{\theta}} \text{bias} + 1)^2}{\boldsymbol{I}} + \text{bias}^2 \leq \mathbb{E}[(\hat{\theta} - \theta)^2]$

Causal Models

Causal scenarios —

- Causal scenario without selection bias: $\mathcal{X}$ affects $\mathcal{Y}$ and there is no selection bias
  - Some features $\mathcal{X}_{\perp Y}$ do not causally affect $\mathcal{Y}$, but are affected by $\mathcal{W}$
  - Some features $\mathcal{X}_{\perp W}$ causally affect $\mathcal{Y}$, but are not affected by $\mathcal{W}$
  - Some features $\mathcal{X}_{W\&Y}$ causally affect $\mathcal{Y}$ and are affected by $\mathcal{W}$ as well as $\mathcal{X}_{\perp Y}$ and $\mathcal{X}_{\perp W}$
- Anti causal scenario: We assume $\mathcal{Y}$ affects $\mathcal{X}$, rather than the other way around
- Causal scenario with selection bias: $\mathcal{X}$ affects $\mathcal{Y}$ and there is a selection bias

Counterfactual invariance —

- Counterfactual invariance: Results of estimator remain consistent across different counterfactual scenarios, i.e. if $\mathcal{Y}$ is affected by $\mathcal{X}$, and $\mathcal{X}$ is affected by $\mathcal{W}$, but $\mathcal{W}$ does not affect $\mathcal{Y}$, our estimator should be invariant to states of $\mathcal{W}$, i.e. $f(\mathcal{X}(\mathcal{W}_1)) = f(\mathcal{X}(\mathcal{W}_2))$
- For counterfactual invariance, the following must hold:
  - Causal scenario without selection bias: $f(\mathcal{X}) \perp \mathcal{W}$, i.e. estimate f only depends on $\mathcal{X}_{\perp W}$
  - Anti causal scenario: $(f(\mathcal{X}) \perp \mathcal{W})|\mathcal{Y}$, i.e. estimate f only depends on $\mathcal{X}_{\perp W}$,

provided $\mathcal{Y}$ is known

  - Causal scenario with selection bias: $(f(\mathcal{X})\perp\mathcal{W})|\mathcal{Y}$ as long as $\mathcal{X}_{\perp Y}$ and $\mathcal{X}_{W\&Y}$ do not influence $\mathcal{Y}$ whatsoever, i.e. $(\mathcal{Y}\perp\mathcal{X})|\mathcal{X}_{\perp W},\mathcal{W}$
- For causal scenario without selection bias we need to show: $\mathcal{X}_{\perp W}\perp\mathcal{W}$
- For anti causal scenario we need to show: $(\mathcal{X}_{\perp W}\perp\mathcal{W})|\mathcal{Y}$
- This can be shown via *d-separation*

D separation —

- Undirected path of n nodes is d-separated, if it contains 3 nodes following any of the following forms and if this form is blocked:
  - Chain structure: $X \to Z \to Y$ or $Y \to Z \to X$ – is blocked, if we condition on Z, i.e. Z is known
  - Fork structure: $X \leftarrow Z \to Y$ – is blocked, if we condition on Z, i.e. Z is known
  - Collider structure: $X \to Z \leftarrow Y$ – is blocked, if we don't condition on Z or any of its descendants
- Random variables X and Y are conditionally independent if each path between them is d-separated

  $\to$ as soon as we have one blocked triple on path, entire path is blocked

  $\to$ as soon as one path is active, we cannot guarantee conditional independence
- For causal scenario without selection bias we can show $\mathcal{X}_{\perp W}\perp\mathcal{W}$ since all paths are blocked
- For anti causal scenario we can show $(\mathcal{X}_{\perp W}\perp\mathcal{W})|\mathcal{Y}$ since all paths are blocked, conditioned on $\mathcal{Y}$, i.e. if $\mathcal{Y}$ is observed