Linear Algebra

Positve (Semi)Definiteness

A positive definite $\iff$ $x^TAx > 0\ \forall x \neq 0$
A positive semidefinite $\iff$ $x^TAx \geq 0\ \forall x$
If A is positive definite:

1. A is invertible
2. $\forall$ eigenvalues λ_i, $\lambda_i > 0$
3. n linearly independent vectors such that $A_{ij} = x_i^T x_j$

If A is positive semidefinite:

1. $A + \gamma I$ is positive definite for any $\gamma > 0$.

Diagonalization: If $n \times n$ matrix A has n linearly independent eigenvectors, $A = PDP^{-1}$. If A is also symmetric, $P^{-1} = P^T$, so $A = PDP^T$.
Trace Properties: $Tr(A) = a_{11} + \cdots + a_{nn} = \sum_{i=1}^n a_{ii}$
$Tr(A+B) = Tr(A) + Tr(B) | Tr(cA) = cTr(A) | Tr(A) = Tr(A^T)$
Cyclic permutations:
$Tr(ABCD) = Tr(BCDA) = Tr(CDAB) = Tr(DABC)$
$Tr(XY^T) = \sum_{i,j} X_{ij}Y_{ij}$
If A is $n \times n$, $Tr(A) = \sum_i \lambda_i$, $det(A) = \prod_i \lambda_i$, $Tr(A^k) = \sum_i \lambda_i^k$
Let x be a scalar. Then $x = Tr(x)$.
Matrix Inversion Lemma
$w = (X^TX + \lambda I)^{-1}X^Ty = X^T(XX^T + \lambda I)^{-1}y$
Rayleigh Quotient: $R(M, x) = \frac{x^TMx}{x^Tx}$

Probability

Expectation: $E[X] = x_1p_1 + \cdots + x_np_n$
(Co)variance
$Cov(X,Y) = E[(X - E[X])(Y - E[Y])^T]$
$Cov(X) = Var(X) = E[(X - E[X])(X - E[X])^T]$
$Var(X) = E[(X - E[X])^2] = E[X^2] - E[X]^2$
$Var(\sum_{i=1}^n X_i) = \sum_{i=1}^n Var(X_i) | Var(nX) = n^2Var(X)$
Bayesian Decision Theory
$P(y|x) = \frac{P(x,y}{P(x)} = P(x|y)\frac{P(y)}{P(x)} = \sum_i P(x|y_i)P(y_i)$
$P(x,y) = P(x|y)P(y) = P(y|x)P(x)$
Decision Boundaries
Let $P(X|w_i) \sim \mathcal{N}(\mu_i, \sigma^2)$, $P(w_1) = P(w_2) = 0.5$. Optimal bayes decision boundary occurs when $P(w_1|x) = P(w_2|x)$ (Pr of being classified in class 1 is equivalent to Pr classified in class 2 for a given x):
$Pr(w_1|x) = Pr(w_2|x) \rightarrow Pr(x|w_1)Pr(w_1) = Pr(x|w_2)Pr(w_2)$
Since $Pr(w_1) = Pr(w_2)$, $Pr(x|w_1) = Pr(x|w_2)$
We know $Pr(x|w_i) \sim \mathcal{N}(\mu_i, \sigma^2)$: $\mathcal{N}(\mu_1, \sigma^2) = \mathcal{N}(\mu_2, \sigma^2)$
Substituting equation for single-variate gaussian and cancelling terms:
$(x - \mu_1)^2 = (x - \mu_2)^2$ so $x = \frac{\mu_1 + \mu_2}{2}$ or in other words the mean of the means. Our decision becomes w_1 if $x < \frac{\mu_1 + \mu_2}{2}$ and w_2 otherwise.
Bayes risk is defined as the probability of misclassification for the Bayes' classifer.
Maximum a Posteriori Estimation
In MAP, we have $P(\theta)$, so we find $P(\theta|x) \propto P(x|\theta)P(\theta)$. The $\theta_{MAP} = argmax_\theta P(x|\theta)P(\theta)$ and can use log for MAP estimation such that $\theta_{MAP} = argmax_\theta \log P(x|\theta) + \log p(\theta)$
Maximum Likelihood Estimation
$\theta_{MLE} = argmax_\theta P_x(x|\theta) = L(\theta)$
$p_{X_1,\ldots,X_n|\theta}(x_1, \ldots, x_n|\theta) = p_{X_1}(X_1|\theta) \ldots p_{X_n}(X_n|\theta)$
Taking the log of the likelihood: $\theta_{MLE} = argmax_\theta \sum_i \log p_{X_i}(x_i|\theta)$
Union Bound $P(\bigcup_i A_i) \leq \sum_i P(A_i)$

Distributions

Single-variate Gaussian: $\frac{1}{\sigma\sqrt{2\pi}}\exp(-\frac{(x-\mu)^2}{2\sigma^2})$
Multi-variate Gaussian with $X \sim \mathcal{N}(\mu, \Sigma)$
$f(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} exp\left(-\frac{1}{2}(x - \mu)^T\Sigma^{-1}(x - \mu)\right)$
Covariance Matrix: $\Sigma = E[(X - \mu)(X - \mu)^T]$; if $\mu = 0$, $\Sigma = XX^T$
Shifting/Centering Distributions:
If $X \sim N(\mu, \Sigma)$, then $AX + b \sim N(A\mu + b, A\Sigma A^T)$
$\implies \Sigma^{-\frac{1}{2}}(X - \mu) \sim N(0, I)$, where $\Sigma^{-\frac{1}{2}} = U\Lambda^{-\frac{1}{2}}$
Covariance Matrix Properties

$x^T\Sigma^{-1}x = ||Ax||_2^2$ for all vectors x when $A = (UQ)^T$ where Q is the diagonal matrix with diagonal values $\frac{1}{\sqrt{(d_{i,i}}}$ so we have
$x^T\Sigma^{-1}x = x^TA^TAx = (Ax)^TAx = ||Ax||_2^2$. This is just the L2 norm of Ax which thus measures squared distance of data vector x from mean.

1. If $||x||_2 = 1$, we can define $||Ux||_2 = 1$ since U is orthonormal (preserving magnitiude). Let $q = Ux$.
$||Ax||_2^2 = x^TA^TAx = x^TUD^{-1}U^Tx = q^TD^{-1}q$ Let x be e_i such that the ith element is 1 and others are 0. Max value of $||Ax||_2^2 = \frac{1}{\lambda_i}$ where λ_i is min eigenvalue and vice versa for min value.
2. If $X_i \perp X_j$, then $cov(X_i, X_j) = 0$. Thus Σ^{-1} is a diagonal matrix of values $\frac{1}{\sigma_i^2}$. Max at $\frac{1}{\sigma_i^2}$ where σ_i^2 is min variance and vice versa for min.

Conclusion: Minimize by choosing vector x as eigenvector corresponding to max eigenvalue or maximium variance if independent.
MLE of Multi-variate Guassian Distribution
For X following the given distribution, we compute the mean and variance estimates using MLE by taking the log-likelihood, and then differentiating by μ and σ, respectively.
$X \sim \mathcal{N}(\mu, \sigma^2 I_{d\times d})$
$\hat{\mu}_{ML} = \frac{\sum_{i=1}^n X_i}{n} = \bar{X}$, $\hat{\sigma^2}_{ML} = \frac{1}{nd}\sum_{i=1}^n (X_i - \hat{\mu}_{ML})^T(X_i - \mu_{\hat{M}L})$
$X \sim \mathcal{N}(\mu, \Lambda)$ where Λ is a diagonal matrix with the values $\sigma_1, \ldots, \sigma_d$.

$$\log l(\mu, \Lambda|X) = -\frac{n}{2}\log 2\pi - \frac{n}{2}\sum_{j=1}^n \log \sigma_j^2 - \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^d \frac{(X_{ij} - \mu_j)^2}{\sigma_j^2}$$

$\mu_{\hat{M}L} = \frac{1}{n}\sum_{i=1}^n X_i$, $\sigma^2_{j,ML}\hat{} = \frac{\sum_{i=1}^n (X_{ij} - \mu_j)^2}{n}$
$X \sim \mathcal{N}(A\mu, \Lambda)$ assuming A is invertible and $Lambda$ is known
$\mu_{\hat{M}}L = A^{-1}\frac{\sum_{i=1}^n X_i}{n}$

Vectors and Matrices

Norms

- $||A||_F^2 = Tr(A^TA)$
- $||x||_2^2 = \sum_{i=1}^n x_i^2 = x^Tx$
- $||x||_1 = \sum_{i=1}^n |x_i|$

Cachy-Schwarz Inequality: $|x^Ty| = ||x||_2||y||_2$
Derivatives

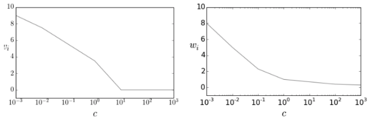
- $\frac{\partial x^Ta}{\partial x} = \frac{\partial a^Tx}{x} = a$
- $\frac{\partial a^TXb}{\partial X} = ab^T$
- $\frac{\partial b^TX^TXc}{\partial X} = X(bc^T + cb^T)$
- $\frac{\partial x^TBx}{\partial x} = (B + B^T)x$
- $\frac{\partial Tr(F(X)}{\partial X} = f(X)^T$
- $\frac{\partial}{\partial x}||x - a||_2 = 2x$
- $\frac{\partial}{\partial X}||X||_F^2 = 2X$

Support Vector Machines

Goal: Maximize the margin of a hyperplane Margin of Hyperplane:
$\frac{min_{1\leq i\leq n} y_i(w^tx)}{||w||_2^2}$

Regression

Linear
Loss Function: $L = ||Xw - Y||_2^2$ with closed-form solution
$w = (X^TX)^{-1}X^Ty$
Loss Function with L2 Regularization: $L = ||Xw - Y||_2^2 + \lambda||w||_2^2$ with closed-form solution $w = (X^TX + \lambda I)^{-1}X^TY$
Loss Function with L1 Regularization: $L = ||Xw - Y||_2^2 + \lambda||w||_1$
How the L1 norm encourages sparsity:
Write loss function in terms of $L(w) = g(y) + \sum_{i=1}^d f(X_i, y, w_i, \lambda)$.
$L(w) = ||Xw - y||_2^2 + \lambda||w||_1 = y^Ty + \sum_{i=1}^d -y^TX_iw_i + \frac{n}{2}w_i^2 + \lambda|w_i|$
If $w_i > 0$, $w_i = \frac{1}{n}(y^TX_i - \lambda)$.
If $w_i < 0$, $w_i = \frac{1}{n}(y^TX_i + \lambda)$.
$w_i = 0$ if $-\lambda \leq y^TX_i \leq \lambda$
Observing regularization paths:



Graph on the left is L1 regularization for sparser values. The graph on the right is L2 regularization, with smoother curve hitting more values.
Writing w as a linear combination of α such that $w = \sum_{i=1}^n \alpha_ix_i$.
$w = X^T(XX^T + \lambda I)^{-1}y$ and we know that $(XX^T + \lambda I)$ is diagonalizable because PD, so equals $U\Lambda U^T$ and thus has inverse $U\Lambda^{-1}U^T$.

$$w = \sum_{i=1}^n \alpha_ix_i$$

$$\alpha_i = \sum_{j=1}^d y_j * u_i\Lambda^{-1}u_j^T$$

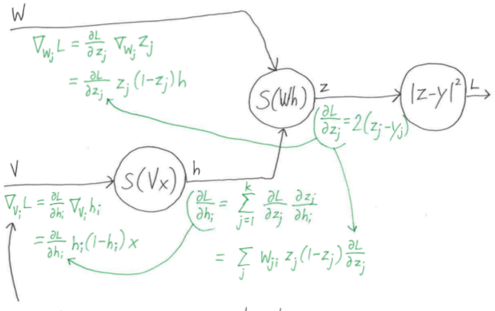
Logistic
Logit Function: $f(x) = \frac{1}{1+exp(-x)}$, $f'(x) = f(x)(1 - f(x))$
$f'(x) = \frac{exp(-x)}{(1+exp(-x))^2} = \frac{exp(-x)+1-1}{(1+exp(-x))^2} = \frac{1+exp(-x)}{(1+exp(-x))^2} - \frac{1}{(1+exp(-x))^2} = \frac{1}{1+exp(-x)} - \frac{1}{(1+exp(-x))^2} = \frac{1}{1+exp(-x)}(1 - \frac{1}{1+exp(-x)} = f(x)(1 - f(x))$
Loss function: $L = -\sum_{i=1}^n y_i \log \mu_i + (1 - y_i)\log \mu_i$ where $\mu_i = \frac{1}{1+exp(-\beta x)}$; $\nabla L = -X^T(Y - \mu)$

Bias-Variance

Given $x_1, \ldots, x_n$, $y = f(x) + \epsilon$, $\hat{(y)} = h(x)$, and $\epsilon \sim \mathcal{N}(0, \sigma^2)$.
$E[(h(x) - y)^2] = E[(h(x) - f(x) + \epsilon)^2] = E[((h(x) - f(x)) + \epsilon)^2] = E[(h(x) - f(x))^2] - 2E[(h(x) - f(x))\epsilon] + E[\epsilon^2]$
We know that $E[\epsilon] = 0$ so $Var(\epsilon) = E[\epsilon^2] + E[\epsilon]^2$ and thus $Var(\epsilon) = E[\epsilon^2]$. Also ϵ is independent of $h(x) - f(x)$ so the middle term is zero.
$E[(h(x) - f(x))^2] + Var(\epsilon) =$
$E[(h(x) - E[h(x)] + E[h(x)] - f(x))^2] + Var(\epsilon) =$
$E[((h(x) - E[h(x)]) + (E[h(x)] - f(x)))^2] + Var(\epsilon) = E[(h(x) - E[h(x)])^2] + 2E[(h(x) - E[h(x)])(E[h(x)] - f(x))] + E[(E[h(x)] - f(x))^2] + Var(\epsilon)$
We know $E[x - \mu] = 0$, so the middle term again disappears.
$E[||h(x) - E[h(x)]||^2] + E[||E[h(x)] - f(x)||^2] + Var(\epsilon)$
We can now define the following:

- Bias: $E[||E[h(x)] - f(x)||^2] = ||E[\hat{\mu}] - \mu||^2$
- Variance: $E[||h(x) - E[h(x)]||^2] = E[||\hat{mu} - E[\hat{\mu}]||^2]$
- Error: $Var(\epsilon)$

Neural Networks


W
$\nabla_{W_j}L = \frac{\partial L}{\partial z_j}\nabla_{W_j}z_j$
$= \frac{\partial L}{\partial z_j} z_j(1-z_j)h$
$S(Wh)$
z
$|z-y|^2$
L
$\frac{\partial L}{\partial z_j} = 2(z_j - y_j)$
V
$S(Vx)$
h
$\nabla_{V_i}L = \frac{\partial L}{\partial h_i}\nabla_{V_i}h_i$
$= \frac{\partial L}{\partial h_i} h_i(1-h_i)x$
$\frac{\partial L}{\partial h_i} = \sum_{j=1}^k \frac{\partial L}{\partial z_j}\frac{\partial z_j}{\partial h_i}$
$= \sum_j W_{ji} z_j(1-z_j)\frac{\partial L}{\partial z_j}$

Compute $\nabla_V L$, $\nabla_W L$ one row at a time.

Softmax: $\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{y_k}}$ for $j = 1, \ldots, k$

Convolutional NN

- Apply filter for example an array $[-1, 0, 1]$.
- Multiply pointwise and sum. Output gets weighted sum.
- Pooling to reduce dimensionality. For example max pooling takes max over every block of 4 pixels.
- Convolving dimension reduction. $n \times m$ filter reduces the dimensions of the height and width of the matrix $d \times d$ matrix to $d - (n-1) \times d - (m-1)$.
- Number of parameters at each level are the parameters in the $n \times n$, times number of matrix-matrix pair, plus a bias unit for each matrix in the output.

Kernels

$$min||Xw - y||_2^2 + \lambda||w||_2^2$$

$w = (X^TX + \lambda I)^{-1}X^Ty = X^T(XX^T + \lambda I)^{-1}y$ and let XX^T define our kernel matrix K.
$\min_w \sum_{i=1}^n loss(w^Tx_i, y_i) + \lambda||w||_2^2$ with $w = \sum_{j=1}^n \alpha_j x_j + v$. We are writing w in terms of linear combinations of α where $\forall i$, $v^Tx_i = 0$.

$$\min_\alpha \sum_{i=1}^n loss(\sum_{j=1}^n \alpha_j x_j^T x_i, y_i) + \lambda||\sum_{j=1}^n \alpha_j x_j||_2^2$$

Now define $K_{ij} = x_i^Tx_j$.

$$\min_\alpha \sum_{i=1}^n loss(\sum_{j=1}^n [K\alpha]_i, y_i) + \lambda\alpha^TK\alpha$$

For predicting new x, $f(x) = w^Tx = \sum_{i=1}^n \alpha_i x_i^T x$, basically computing the similarity between every sample point x_i to x. You can replace x_i^Tx with any appropriate kernel function $k(x, z)$.

Kernel Functions

1. linear: $k(x, z) = x^Tz$
2. quadratic: $k(x, z) = (1 + x^Tz)^2$
3. gaussian: $k(x, z) = exp(\gamma||x - z||^2)$

With matrix notation, we can write the kernel function as $min||K\alpha - y||^2 + \lambda\alpha^TK\alpha$ with the solution $\alpha = (K + \lambda I)^{-1}y$.
We can also define $K = \phi(x)\phi(x)^T$ where $\phi(x)$ is some feature mapping for example $\phi(x) = \begin{bmatrix}1 & \sqrt{2} & x^2\end{bmatrix}^T$

Nearest Neighbors

Find k-closest training points and classify as most likely label by some voting scheme such as mean or median.

Decision Trees

Entropy is the expected value of surprise: $e = -\sum_{i=1}^n p_i \log_b(p_i)$
If $p_1 = 1$ and $p_2 = 0$, then
$e = -(p_1 \log_2 p_1 + p_2 \log_2 p_2) = -(1 * \log_2 1 + 0) = 0.$
If $p_1 = p_2 = 0.5$ and $p_0 = 0$, then
$e = -(p_1 \log_2 p_1 + p_2 \log_2 p_2) = -(0.5 * \log_2 0.5 + 0.5 * \log_2 0.5) = 1.$
Maximize information gain for classification. With entropy H at current node with $n_1 + n_2$ samples: $maxH - \frac{n_1H_- + n_2H_+}{n_1+n_2}$.

- Info gain at root is not necessarily greater than info gain at a lower level; consider the XOR function.
- The same feature can be split on twice.
- Entropy is always nonnegative.

Principle Components Analysis

Singular Value Decomposition

$$X = USV^T = \sum_{i=1}^d \sigma_i u_i v_i^T$$

1. U is $d \times d$ with the left singular vectors
2. S is $d \times d$ with the singular values on the diagonal.
3. V is $n \times d$ with the right singular vectors
4. $U^TU = I$ and $V^TV = I$ since both U and V contain orthogonal vectors.
5. $S = Diag(\sigma_i)$ where singular values ordered from greatest to least $\sigma_1 \geq \cdots \geq 0$.

The singular values of X are the square roots of the eigenvalues for X^TX or XX^T. The left singular vectors are the eigenvectors of XX^T and the right singular vectors are the eigenvectors of X^TX.

Norms in terms of SVD

$||A||_F^2 = Trace(A^TA) = Trace(VS^TU^TUSV^T) = Trace(VSSV^T) = Trace(V^TVSS) = Trace(SS) = \sum_i \sigma_i^2$
$||A||_2^2 = sup_{||x||=1}||Ax||_2 = sup_{||x||=1}x^TA^TAx$ We see that we are just finding the suprenum of the Raleigh quotient, which is maximized with the largest eigenvalue of A^TA which is equivalent to σ_1^2.

PCA Algorithm

Center X, compute SVD of X, then return $\hat{X} = S_rV_r^T, U_r, \mu_x$

Latent Factor Analysis

Goal: Factor X in AB^T. $min_{A,B}||X - AB^T||_F^2$. The solution is $A = U_rS_r^{\frac{1}{2}}$ and $B = V_rS_r^{\frac{1}{2}}$.

Clustering

K-means Clustering

$$J = \sum_{j=1}^k \sum_{i=1} n||x_i^{(j)} - c_j||^2$$

For each cluster, for each point, compute distance to the assigned mean. Algorithm: Fix points, update means to be mean of assigned points. Fix means, reassign points to new closest mean. Repeat until convergence.

Spectral Clustering

Weighted undirected graph $G = (V, E)$ where edge (i, j) represents similarities between points x_i and x_j. Cut graph into 2 pieces, cutting minimum edge weight each time: $min\frac{Cut(G_1,G_2}{Mass(G_1)Mass(G)}$ We define the Laplacian as:

$$L_{ij} = -\begin{cases}\sum_j a_{ij} & \text{if i=j} \\ -a_{ij} & \text{o.w.}\end{cases}$$

Let v be a vector indicating whether v_i is in graph V_1 or V_2.
$Cut(V_1, V_2) = \sum_{i,j} \frac{1}{4}(v_i - v_j)^2 w_{ij} =$
$\frac{1}{4}\sum_{i,j} w_{ij}v_i^2 + w_{ij}v_j^2 - 2v_iv_jw_{ij} = \frac{1}{4}v^TLv$. We can approximate this with the following: $min\frac{1}{4}v^TLv$ such that $||v||^2 = n$ and $1^Tv = 0$. The solution here is the second smallest eigenvalue. Without constraint $1^Tv = 0$, the solution would be smallest eigenvalue because it is a Rayleigh quotient. 1^Tv indicates you are orthogonal to the all 1's vector which corresponds to the smallest eigenvalue, so we take the second smallest.