1 Regression

1.1 Linear Regression

Multiple

$f(\mathbf{x_n}) := w_0 + \sum_{j=1}^D w_j x_{nj} = \tilde{\mathbf{x}}_n^T \mathbf{w}$

If $D > N$ the task is under-determined (more dimensions than data) → regularisation.

2 Cost functions

MSE = $\frac{1}{N}\sum_{n=1}^N [y_n - f(\mathbf{x_n})]^2$

MAE = $\frac{1}{N}\sum_{n=1}^N |y_n - f(\mathbf{x_n})|$

2.1 Convexity

$f(\lambda \mathbf{u} + (1-\lambda)\mathbf{v}) \le \lambda f(\mathbf{u}) + (1-\lambda) f(\mathbf{v})$ with $\lambda\in[0;1]$. A strictly convex function has a unique global minimum w^*. A function must always lie above its linearisation: $\mathcal{L}(u) \ge \mathcal{L}(w) + \nabla\mathcal{L}(w)^T(u-w) \forall u,w$.

A set is convex iff line segment between any two points of $\mathcal{C}$ lies in $\mathcal{C}$: $\theta u + (1-\theta)v \in \mathcal{C}$

3 Optimisation

Gradient $\nabla\mathcal{L} := \left[\frac{\partial \mathcal{L}(\mathbf{w})}{\partial w_1} \quad \dots \quad \frac{\partial\mathcal{L}(\mathbf{w})}{\partial w_D}\right]$

3.1 Gradient descent

$\mathbf{w^{(t+1)}} = \mathbf{w^{(t)}} - \gamma\nabla\mathcal{L}(\mathbf{w^{(t)}})$. Very sensitive to ill-conditioning.

GD - Linear Reg

$\mathcal{L}(\mathbf{w}) = \frac{1}{2N}(\mathbf{y}-X\mathbf{w})^T(\mathbf{y}-X\mathbf{w}) \to \nabla\mathcal{L}(\mathbf{w}) = -\frac{1}{N}X^T(\mathbf{y}-X\mathbf{w})$. Cost: $O_{err} = 2ND + N$ and $O_w = 2ND + D$.

3.2 SGD

$\mathcal{L} = \frac{1}{N}\sum \mathcal{L}_n(\mathbf{w})$ with update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \gamma\nabla\mathcal{L}_n(\mathbf{w}^{(t)})$.

3.3 Mini-batch SGD

$\mathbf{g} = \frac{1}{|B|}\sum_{n\in B}\nabla\mathcal{L}_n(\mathbf{w}^{(t)})$ with update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \gamma\mathbf{g}$.

3.4 Subgradient at w

$\mathbf{g}\in\mathbb{R}^D$ such that $\mathcal{L}(u) \ge \mathcal{L}(w) + \mathbf{g}^T(u-w)$. Example subgradient for MAE: $h(e) = |e| \to g(e) = sgn(e)$ if $e\neq 0, \lambda\in[-1;1]$ otherwise.

We get the gradient:

$\nabla\mathcal{L}_{MAE} = -\frac{1}{N}\sum_n sgn(y - f(x_n))\nabla f(x_n)$.

3.5 Projected SGD

$\mathbf{w}^{(t+1)} = \mathcal{P}_\mathcal{C}[\mathbf{w}^{(t)} - \gamma\nabla\mathcal{L}(\mathbf{w}^{(t)})]$

3.6 Newton's method

Second order (more expensive $O(ND^2 + D^3)$ but faster convergence).

$w^{(t+1)} = w^{(t)} - \gamma^{(t)}(H^{(t)})^{-1}\nabla\mathcal{L}(w^{(t)})$

3.7 Optimality conditions

Necessary: $\nabla\mathcal{L}(\mathbf{w}^*) = 0$ Sufficient: Hessian PSD $\mathbf{H}(\mathbf{w}^*) := \frac{\partial^2\mathcal{L}(\mathbf{w}^*)}{\partial w\partial w^T}$

4 Least Squares

4.1 Normal Equation

$X^T(\mathbf{y} - X\mathbf{w}) = 0 \Rightarrow \mathbf{w}^* = (X^TX)^{-1}X^T\mathbf{y}$ and $\hat{\mathbf{y}}_\mathbf{m} = \mathbf{x_m}^T\mathbf{w}^*$

Graham matrix invertible iff $rank(X) = D$ (use SVD $X = USV^T \in \mathbb{R}^{N\times D}$ if this is not the case to get pseudo-inverse $\mathbf{w}^* = V\tilde{S}U^T$ with $\tilde{S}$ pseudo-inverse of S).

5 Likelihood

Probabilistic model $y_n = \mathbf{x_n}^T\mathbf{w} + \epsilon_n$. Probability of observing the data given a set of parameters and inputs: $p(\mathbf{y}|X,\mathbf{w}) = \prod p(y_n|\mathbf{x_n},\mathbf{w}) = \prod\mathcal{N}(y_n|\mathbf{x_n}^T\mathbf{w},\sigma^2)$

Best model maximises log-likelihood $\mathcal{L}_{LL} = -\frac{1}{2\sigma^2}\sum(y_n - x_n^Tw)^2 + cst$.

6 Regularisation

6.1 Ridge Regression

$\mathcal{L}(\mathbf{w}) = \frac12(\mathbf{y}-X\mathbf{w})^T(\mathbf{y}-X\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2 \to$

$\mathbf{w}^*_{\mathbf{ridge}} = (XX^T + \lambda I_D)^{-1}X^T\mathbf{y}$

$= X^T(XX^T+\lambda I_N)^{-1}\mathbf{y}$

Can be considered a MAP estimator: $\mathbf{w}^*_{\mathbf{ridge}} = argmin_w - log(p(w|X,y))$

6.2 Lasso

Sparse solution. $\mathcal{L}(w) = \frac{1}{2N}(y - Xw)^T(y-Xw) + \lambda\|w\|_1$

7 Model Selection

7.1 Bias-Variance decomposition

Small dimensions: large bias, small variance. Large dimensions: small bias, large variance. Error for the val set compared to the emp distr of the data $\propto \sqrt{ln(|\Omega|)}/\sqrt{|V|}$

8 Classification

8.1 Optimal

$\hat{y}(\mathbf{x}) = argmax_{y\in\mathcal{Y}}p(y|\mathbf{x})$

8.2 Logistic regression

$\sigma(z) = \frac{e^z}{1+e^z}$ to limit the predicted values $y\in[0;1]$ ($p(1|\mathbf{x}) = \sigma(\mathbf{x}^T\mathbf{w})$ and $p(0|\mathbf{x}) = 1-\sigma(\mathbf{x}^T\mathbf{w})$). Decision wrt 0.5.

Likelihood

$p(y|X,w) = \prod p(y_n|x_n)$

$= p(0|x_n)\dots p(K|x_n)$

$= \prod_k\prod_n[p(y_n = k|x_n,w)]^{\tilde{y}_{nk}}$

where $\tilde{y}_{nk} = 1$ if $y_n = k$.

For binary classification

$p(y|X,w) = \prod p(y_n|x_n)$

$= \prod_{n:y_n=0} p(0|x_n)\prod_{n:y_n=1}p(1|x_n)$

$= \prod_n\sigma(x_n^Tw)^{y_n}[1-\sigma(x_n^Tw)]^{1-y_n}$

Loss

$\mathcal{L}(w) = \sum_{n=1}^N ln(1+exp(x_n^Tw)) - y_nx_n^Tw$ which is convex in w.

Gradient

$\nabla\mathcal{L}(w) = \sum_{n=1}^N x_n(\sigma(x_n^Tw) - y_n) = X^T[\sigma(Xw) - y]$ (no closed form solution).

Hessian $H(w) = X^TSX$

with $S_{nn} = \sigma(x_n^Tw)[1-\sigma(x_n^Tw)]$

8.3 Exponential family

General form

$p(y|\eta) = h(y)exp[\eta^T\psi(y) - A(\eta)]$

Cumulant

$A(\eta) = ln[\int_y h(y)exp[\eta^T\psi(y)]dy]$

$\nabla A(\eta) = \mathbb{E}[\psi(y)] = g^{-1}(\eta)$

$\nabla^2 A(\eta) = \mathbb{E}[\psi\psi^T] - \mathbb{E}[\psi]\mathbb{E}[\psi^T]$

Link function

$\eta = g(\mu) \Leftrightarrow \mu = g^{-1}(\eta)$

$\eta_{gaussian} = (\mu/\sigma^2, -1/2\sigma^2)$; $\eta_{poisson} = ln(\mu)$; $\eta_{bernoulli} = ln(\mu/1-\mu)$

$\eta_{general} = g^{-1}(\frac1N\sum_{n=1}^N\psi(y_n))$

$\nabla\mathcal{L}(w)X^T[g^{-1}(Xw) - \psi(y)] = 0$

8.4 Nearest Neighbor Models

Performs best in low dimensions.

8.4.1 k-NN

$f_{S_{t,k}}(x) = \frac1k\sum_{n:x_n\in ngbh_{S_{t,k}(x)}} y_n$ Pick odd k so there is a clear winner. Large $k\to$ large bias small variance (inv.)

8.4.2 Error bound

$\mathbb{E}[\mathcal{L}_{S_t}] \le 2\mathcal{L}_{f^*} + 4c\sqrt{d}N^{-1/d+1}$

8.5 Support Vector Machines (SVM)

Logistic regression with hinge loss: $min_w\sum_{n=1}^N[1-y_nx_n^Tw]_+ + \frac\lambda2\|w\|^2$ where $y\in[-1;1]$ the label and $hinge(x) = max\{0,x\}$. Convex but not differentiable so need subgradient.

Duality: $\mathcal{L}(w) = max_\alpha G(w,\alpha)$. For SVM $min_w max_{\alpha\in[0,1]^N}\sum\alpha_n(1-y_nx_n^Tw) + \frac12\|w\|^2$ differentiable and convex.

Can switch max and min when convex in w and concave in α. Simpler form:

$w(\alpha) = \frac1\lambda\sum\alpha_ny_nx_n = \frac1\lambda X^Tdiag(y)\alpha$ which yields the optimisation problem:

$max_{\alpha\in[0,1]^N}\alpha^T\mathbf{1} - \frac{1}{2\lambda}\alpha^TYXX^TY\alpha$

The solution is sparse (α_n is the slope of the lines that are lower bounds to the hinge loss).

8.6 Kernel Ridge Regression

From duality $w^* := X^T\alpha^*$ where $\alpha^* := (K+\lambda I_N)^{-1}y$ and $K = XX^T = \phi^T(x)\phi(x) = \kappa(x,x')$ (needs to be PSD and symmetric).

9 Unsupervised Learning

9.1 K-means clustering

$min\,\mathcal{L}(z,\mu) = \sum_n^N\sum_k^K z_{nk}\|x_n - \mu_k\|_2^2$ with $z_{nk}\in\{0,1\}$ (unique assignments: $\sum_k z_{nk} = 1$).

Algorithm (Coordinate Descent)

1. $\forall n, z_n = \begin{cases}1 \text{ if } k = argmin_j\|x_n-\mu_k\|^2\\ 0 \text{ otherwise}\end{cases}$

2. $\forall k$ compute $\mu_k = \sum_n z_{nk}x_n/\sum_n z_{nk}$

Pb:cost,spher+hard clusters

Probabilistic model

$p(X|\mu,z) = \prod_n\mathcal{N}(x_n|\mu_k,I)$

$= \prod_n\prod_k\mathcal{N}(x_n|\mu_k,I)^{z_{nk}}$

9.2 Gaussian Mixture Models

$p(X|\mu,z) = \prod_n(x_n|z_n,\mu_k,\Sigma_k)p(z_n|\pi) = \prod_n\prod_k[\mathcal{N}(x_n|\mu_k,\Sigma_k)]^{z_{nk}}\prod_k[\pi_k]^{z_{nk}}$

where $pi_k = p(z_n = k)$

Marginal likelihood: z_n latent variables => factored out of likelihood $p(x_n|\theta) = \sum\pi_k\mathcal{N}(x_n|\mu_k,\Sigma_k)$.

nb params $O(N)$ to $O(D^2K)$.

9.3 EM

9.3.1 GMM

Intialize $\mu^{(1)},\Sigma^{(1)},\pi^{(1)}$.

1. E-step: Compute the assignments. $q_{kn}^{(t)} := \frac{\pi_k^{(t)}\mathcal{N}(x_n|\mu_k^{(t)},\Sigma_k^{(t)})}{\sum_k^K\pi_k^{(t)}\mathcal{N}(x_n|\mu_k^{(t)},\Sigma_k^{(t)})}$ 2. Compute Marginal Likelihood 3. M-step: Update

$\mu^{(t+1)} = \frac{\sum_n q_{kn}^{(t)}x_n}{\sum_n q_{kn}^{(t)}}$ $\pi^{(t+1)} = \frac1N\sum_n q_{kn}^{(t)}$

$\Sigma^{(t+1)} = \frac{\sum_n q_{kn}^{(t)}(x_n-\mu^{(t+1)})(x_n-\mu^{(t+1)})^T}{\sum_n q_{kn}^{(t)}}$

9.3.2 General

$\theta^{(t+1)} := argmax_\theta\sum_n\mathbb{E}_{p(z_n|x_n,\theta^{(t)})}[log\,p(x_n,z_n|\theta)]$

10 Matrix Factorisations

10.1 Prediction

Find $\mathbf{X}\approx\mathbf{W}\mathbf{Z}^T$ where $\mathbf{W}\in\mathbb{R}^{D\times K}$ and $\mathbf{Z}\in\mathbb{R}^{N\times K}$ with $K << D,N$. Large $K\to$ overfitting. If $K\ge max\{D,N\}$ trivial solution ($W = \mathbf{1}_D$ or $Z = \mathbf{1}_N$).

Quality of reconstruction (not jointly convex nor identifiable):

$\mathcal{L}(\mathbf{W},\mathbf{Z}) := \frac12\sum_{(d,n)\in\Omega}[x_{dn} - (\mathbf{W}\mathbf{Z}^T)_{dn}]^2$

$= \sum_{(d,n)\in\Omega} f_{dn}(w,z)$

Regulariser: $\Omega(W,Z) = \frac{\lambda_w}{2}\|\mathbf{W}\|^2_{Frob} + \frac{\lambda_z}{2}\|\mathbf{Z}\|^2_{Frob}$

Optimisation with SGD (compute ∇_w for a fixed user d' and ∇_z for a fixed item n').

ALS (assume no missing ratings):

$\mathbf{Z}_*^T = (\mathbf{W}^T\mathbf{W} + \lambda_zI_K)^{-1}\mathbf{W}^T\mathbf{X}$

$\mathbf{W}_*^T = (\mathbf{Z}^T\mathbf{Z}+\lambda_wI_K)^{-1}\mathbf{Z}\mathbf{X}^T$

11 Dimensionality reduction

11.1 SVD

$\mathbf{X} = \mathbf{USV}^T$, with $\mathbf{X}: D\times N$, $\mathbf{U}: D\times D$ orthonormal, $\mathbf{V}: N\times N$ orthonormal, $\mathbf{S}: D\times N$ diagonal PSD, values in descending order ($s_1\ge\dots\ge s_D\ge 0$).

Reconstruction

$\|\mathbf{X}-\hat{\mathbf{X}}\|_F^2 \ge \|\mathbf{X} - \mathbf{U}_K\mathbf{U}_K^T\mathbf{X}\|_F^2 = \sum_{i\ge K+1}s_i^2$

∀ rank-K matrix $\hat{\mathbf{X}}$ (i.e. *we should compress the data by projecting it onto these left singular vectors.*)

Truncated SVD: $\mathbf{U}_K\mathbf{U}_K^T\mathbf{X} = \mathbf{US}_K\mathbf{V}^T$

Application to MF: $\mathbf{U} = \mathbf{W}$, $\mathbf{SV}^T = \mathbf{Z}^T$. Rec. limited by the rank-K of W,Z.

11.2 PCA

Decorrelate the data. Empirical cov before: $N\mathbf{K} = \mathbf{XX}^T = \mathbf{US}_D^2\mathbf{U}^T$. After $\tilde{\mathbf{X}} = \mathbf{U}^T\mathbf{X}$: $N\tilde{\mathbf{K}} = \tilde{\mathbf{X}}\tilde{\mathbf{X}}^T = \mathbf{S}_D^2$ (the components are uncorrelated).

Pitfalls: not invariant under scalings.

12 Neural Networks

The output at the node j in layer l is $x_j^{(l)} = \phi\left(\sum_i w_{i,j}^{(l)}x_i^{(l-1)} + b_j^{(l)}\right)$

12.1 Representation power

Error bound $\le\frac{(2Cr)^2}{n}$ where C is the smoothness bound, n the number of nodes. We can approximate any sufficiently smooth 2D function on bounded domain (*on average* with σ activation, *pointwise* with ReLU).

12.2 Learning

Problem is not convex but SGD is stable. Backpropagation: Let $\mathcal{L}_n = (y_n - f^{(L+1)} \circ \cdots \circ f^{(1)}(\mathbf{x}_n^{(0)}))^2$.

Forward pass

$\mathbf{x}^{(0)} = \mathbf{x}_n$. For $l = 1, \ldots, L+1$

$\mathbf{z}^{(l)} = (\mathbf{W}^{(l)})^T \mathbf{x}^{(l-1)} + \mathbf{b}^{(l)}$, $\mathbf{x}^{(l)} = \phi(\mathbf{z}^{(l)})$

Backward pass

$\delta^{(L+1)} = -2(y_n - \mathbf{x}^{(L+1)})\phi'(\mathbf{z}^{(L+1)})$ and $\forall l : \delta^{(l)} = (\mathbf{W}^{(l+1)}\delta^{(l+1)}) \odot \phi'(\mathbf{z}^{(l)})$

Final pass

$\frac{\partial L_n}{\partial w_{i,j}^{(l)}} = \delta_j^{(l)} \mathbf{x}_i^{(l-1)}$, $\frac{\partial L_n}{\partial b_j^{(l)}} = \delta_j^{(l)}$

12.3 Activations

sigmoid $\phi(x) = 1 - \sigma(x)$, tanh $\frac{e^x + e^{-x}}{e^x + e^{-x}} = 2\phi(2x) - 1$, ReLU, Leaky ReLU ($max\{\alpha x, x\}$).

12.4 Convolutional Neural Nets

Convolution with filter f: $x^{(1)}[n,m] = \sum_{k,l} f[k,l] x^{(0)}[n-k, m-l]$. Filter is local so no need for fully connected layers. We can use same filter at every position: *weight sharing*. Learning: run backprop by computing different weights, then sum the gradients of shared weights.

12.5 Overfitting

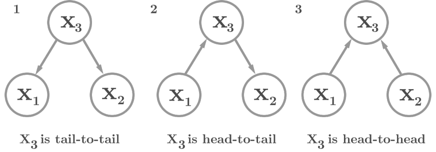
Adding regularisation is equivalent to weight decay (by $(1-\eta\lambda)$). Can also use dataset augmentation, dropout.

13 Graphical Models

13.1 Bayes Nets

$p(X_1, \ldots, X_D) = p(X_1)p(X_2|X_1)\ldots p(X_D|X_1, \ldots, X_{D-1})$. One node is a random variable, directed edge from X_j to X_i if X_j appears in the conditioning $p(X_i|\ldots, X_j, \ldots)$. The graph must be *acyclic*.

Conditional independence: $p(X,Y) = p(X)p(Y)$ or given Z $p(X,Y|Z) = p(X|Z)p(Y|Z)$.



1. $p(x_1, x_2, x_3) = p(x_3)p(x_1|x_3)p(x_2|x_3)$: x_1 and x_2 indep. given x_3
2. $p = p(x_1)p(x_3|x_1)p(x_2|x_3)$: id.
3. $p = p(x_1)p(x_2)p(x_3|x_1, x_2)$: x_1 and x_2 **not** indep. given x_3

$X \to Y$ path blocked by Z if it contains a variable such that either 1. variable is in Z and it is head-to-tail or tail-to-tail 2. node is head-to-head and neither this node nor any of its descendants are in Z.
X and Y are D-sep. by Z iff every path $X \to Y$ is blocked by Z.
X conditionally indep. of Y conditioned on the Z if X and Y are D-sep. by Z. Indep. is symmetric.

14 Quick maff

Chain rule $h = f(g(w)) \to \partial h(w) = \partial f(g(w)) \nabla g(w)$

Gaussian $\mathcal{N}(y|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{(y-\mu)^2}{2\sigma^2})$

Multivariate Gaussian $\mathcal{N}(y|\mu, \sigma^2) = \frac{1}{\sqrt{(2\pi)^D det(\Sigma)}} exp(-\frac{1}{2}(y-\mu)^T \Sigma^{-1}(y-\mu))$

Bayes rule $p(x|y) = \frac{p(y|x)p(x)}{p(y)}$

Logit $\sigma(x) = \frac{\partial ln[1+e^x]}{\partial x}$

Naming Joint distribution $p(x,y) = p(x|y)p(y) = p(y|x)p(x)$ where

- $p(x|y)$ or $p(y|X,w)$ $\to$ likelihood
- $p(y)$ or $p(w) \to$ prior
- $p(y|x) \to$ posterior
- $p(x) \to$ marginal likelihood
- $p(w|y,X) \to$ MAP estimator

Marginal Likelihood

$p(\mathbb{X}|\alpha) = \int_\theta p(\mathbb{X}|\theta)p(\theta|\alpha)\ d\theta$

$p(X = x) = \sum_y p(X = x, Y = y) = \sum_y p(X = x \mid Y = y)p(Y = y)$

Posterior probability $\propto$ Likelihood $\times$ Prior

Maximising over a Gaussian is equivalent to minimising MSE: $\beta^*_{MAP} = argmax_\beta p(y|X,\beta)p(\beta) \Leftrightarrow \beta^* = argmin_\beta \mathcal{L}(\beta)$

Identifiable model $\theta_1 = \theta_2 \to P_{\theta_1} = P_{\theta_2}$

14.1 Algebra

$(PQ + I_N)^{-1}P = P(QP + I_M)^{-1}$

$\sum_n (y_n - \beta^T \mathbf{x_n})^2 = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta)$

$\sum_j \beta^2 = \beta^T \beta$

Unitary / orthogonal: $\mathbf{U}\mathbf{U}^T = \mathbf{U}^T\mathbf{U} = \mathbf{I}$ and $\mathbf{U}^T = \mathbf{U}^{-1}$. Rotation matrix (preserves length of vector).

Jensen's inequality: $log(\sum a) \geq \sum q log(\frac{a}{q})$

15 Mock Exam Notes

15.1 Normal equation

Unique if convex.

$\frac{1}{\sigma_k^2} X(X^T w_k - y_k) + w_k = 0 \Leftrightarrow$

$w_k^* = (\frac{1}{\sigma_k^2} X X^T + I_D)^{-1} \frac{1}{\sigma_k^2} X y_k$

15.2 MAP solution

$\mathcal{L}(w) = \sum_k \sum_n \frac{1}{2\sigma_k^2}(y_{nk} - x_n^T w_k)^2 + \frac{1}{2}\sum_k \|w_k\|_2^2 \to$ Likelihood $p(y|X,w) = \prod_n \prod_k \mathcal{N}(y_{nk}|w_k^T x_n, \sigma_k^2)$ and prior $p(w) = \prod_k \mathcal{N}(w_k|0, I_D)$

15.3 Convexity

$ln[\sum_k^K e^{t_k}]$ is convex. Linear sum of parameters is convex.

15.4 Deriving marginal distribution

$p(y_n|x_n, r_n = k, \beta) = \mathcal{N}(y_n|\beta_k^T \tilde{x}_n, 1)$

Assume r_n follows a multinomial $p(r_n = k|\pi)$. Derive the marginal $p(y_n|x_n, \beta, \pi)$. $p(y_n|x_n, r_n = k, \beta) = \sum_k^K p(y_n, r_n = k|x_n, \beta, \pi) = \sum_k^K p(y_n|r_n = k, x_n, \beta, \pi) \cdot \pi_k = \sum_k^K \mathcal{N}(y_n|\beta_k^T \tilde{x}_n, \sigma^2) \cdot \pi_k$

15.5 MF

$\hat{r}_{um} = \langle \mathbf{v}_u, \mathbf{w}_m \rangle + b_u + b_m$ $\mathcal{L} = \frac{1}{2}\sum_{u\ m}(\hat{r}_{um} - r_{um}) + \frac{\lambda}{2}\left[\sum_u (b_u^2 + \|\mathbf{v}_u\|^2) + \sum_m (b_m^2 + \|\mathbf{w}_m\|^2)\right]$. The optimal value for b_u for a particular user u' : $\sum_{u'\ m}(\hat{r}_{u'm} - r_{u'm}) + \lambda b_{u'} = 0$.

Problem jointly convex? Compute $H(\hat{r}(v,w)) = \begin{bmatrix} 2w^2 & 4vw - 2r \\ 4vw - 2r & 2v^2 \end{bmatrix}$ which is not PSD in general.

16 Multiple Choice Notes

16.1 True statements

- Regularisation term sometimes renders the min. problem into a strictly concave/convex problem.
- k-NN can be applied even if the data cannot be linearly separated.
- $max\{0, x\} = max_{\alpha \in [0,1]} \alpha x$
- $min\{0, x\} = min_{\alpha \in [0,1]} \alpha x$
- $g(x) = min_y f(x,y) \Rightarrow g(x) \leq f(x,y)$
- $max_x g(x) \leq max_x f(x,y)$
- $max_x min_y f(x,y) \leq min_y max_x f(x,y)$
- $\nabla_W(\mathbf{x}^T \mathbf{W} \mathbf{x}) = \mathbf{x}\mathbf{x}^T$
- $\nabla_x(\mathbf{x}^T \mathbf{W} \mathbf{x}) = (\mathbf{W} + \mathbf{W}^T)\mathbf{x}$
- K-means: optimal cluster (resp. centers) init $\to$ one step optimal representation points (resp. clusters).
- Logistic loss is typically preferred over L_2 loss in classification tasks.
- For optimizing a MF of a $D \times N$ matrix, for large D, N : *per iteration, ALS has an increased computational cost over SGD* and *per iteration, SGD cost is independent of D, N.*
- The complexity of backprop for a nn with L layers and K nodes/layer is $O(K^2 L)$
- CNN where the data is laid out in a one-dimensional fashion and the filter/kernel has M non-zero terms. Ignoring the bias terms, there are M parameters.

16.2 Convex functions

- $f(x) = x^\alpha, x \in \mathbb{R}^+, \forall \alpha \geq 1 or \leq 0$
- $f(x) = -x^3, x \in [-1, 0]$
- $f(x) = e^{ax}, \forall x, a \in \mathbb{R}$
- $f(x) = ln(1/x), x \in \mathbb{R}^+$
- $f(x) = g(h(x)), x \in \mathbb{R}, g, h$ convex and increasing over $\mathbb{R}$
- $f(x) = ax + b, x \in \mathbb{R}, \forall a, b \in \mathbb{R}$
- $f(x) = |x|^p, x \in \mathbb{R}, p \geq 1$
- $f(x) = x log(x), x \in \mathbb{R}^+$

16.3 Non-convex functions

- $f(x) = x^3, x \in [-1, 1]$
- $f(x) = e^{-x/2}, x \in \mathbb{R}$
- $\sum \mathcal{N}$
- $sin(x) \forall x \in \mathbb{R}$

17 Mock2014

17.1 Weighted LS

$\mathcal{L}(\beta) = 1/2 \sum_n w_n (y_n - \beta^T \tilde{\mathbf{x}}_n)^2$.

$\partial \mathcal{L}(\beta) = \sum_n w_n (y_n - \beta^T \tilde{\mathbf{x}}_n)\tilde{\mathbf{x}}_n = -\tilde{X}^T W \mathbf{y} + \tilde{X}^T W \tilde{X} \mathbf{B} = 0$.

$w_n > 0 \to$ W pos def $\to \tilde{X}^T W \tilde{X}$ invertible $\to$ unique sol $\beta^* = (\to \tilde{X}^T W \tilde{X})^{-1} \tilde{X}^T W \mathbf{y}$.

prob model : $p(\mathbf{y}|X, \beta) = \prod_n \mathcal{N}(y_n|\beta^T \tilde{\mathbf{x}}_n, 1/w_n)$.

17.2 Multiclass class

$\eta_{nk} = \tilde{\mathbf{x}}_n^T \beta_k$. $p(y_n = k|\mathbf{x}_n, \beta) = \frac{e^{\eta_{nk}}}{\sum_j x^{\eta_{nj}}}$. $p(\mathbf{y}|X, \beta) = \prod p(y_n = k|\mathbf{x}_n, \beta)$. $\tilde{\mathbf{y}}_{nk} = 1$ if $y_n = k$ and 0 else. log-lik : $\log p(\mathbf{y}|X, \beta) = \log \prod_k \prod_n [p(y_n = k|\mathbf{x}_n, \beta)]^{\tilde{\mathbf{y}}_{nk}} =$

17.3 Subgradients

$\mathcal{L}(\mathbf{w}) = MAE(\mathbf{w}) = 1/N \sum_n |y_n - f(\mathbf{w}, \mathbf{w_n})|$. Use chain rule with subgradient $h(x) = sgn(x)$. $\nabla \mathcal{L}(\mathbf{w}) = -1/N \sum_n h(y_n - f(\mathbf{w})) * \nabla f(\mathbf{w}, \mathbf{x_n})$. Then update weights.

17.4 K-means clustering + reg

Change rule for z_{nk} with "...+ $\|\mathbf{u_k}\|$". Derive cost function w.r. to $\mathbf{u_k}$ and =0 to find optimal centers

17.5 Multiple output reg

x_n has dim D but now y_n has dim K. $\mathcal{L}(\mathbf{W}) = \sum_k \sum_n 1/2\sigma_k^2 (y_{nk} - \mathbf{x}_n^T \mathbf{w})^2 + 1/2\sigma_0^2 \sum_k \|\mathbf{w}_k\|^2$. Derive w.r. to a $\mathbf{w}_k$ to get optimal weights : $1/\sigma_k^2 X^T(X\mathbf{w}_k - \mathbf{y}_k) + 1/\sigma_0^2 \mathbf{w}_k = 0$. Pb is convex in W. $\mathbf{w}_k^* = (1/\sigma_k^2 X^T X + 1/\sigma_0^2 I_D)^{-1} 1/\sigma_k^2 X^T \mathbf{y}_k$.

Prob model (posterior) same answer as **15.2** but with $1/2\sigma_0^2 I_D$ for the prior

17.6 Kernels

Prove that sym $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j$ and that pos sym def $t^T K t = \sum_i \sum_j K_{ij} t_i t_j \geq 0 \forall t$

17.7 Mixture of lin reg

$p(y_n|\mathbf{x}_n, r_n = k, \beta) = \mathcal{N}(y_n|\beta_k^T \tilde{\mathbf{x}}_n, 1)$.

We define $\mathbf{r}_{nk}$ like $\mathbf{y}_{nk}$ in **17.2**

Likelihood : $p(y_n|\mathbf{x}_n, \beta, \mathbf{r}_n) = \prod_k [\mathcal{N}(y_n|\beta_k^T \tilde{\mathbf{x}}_n, \sigma^2]^{r_{nk}}$.

LL : $p(\mathbf{y}|X, \beta, \mathbf{r}) = \prod_n \prod_k [\mathcal{N}(y_n|\beta_k^T \tilde{\mathbf{x}}_n, \sigma^2]^{r_{nk}}$. For $p(r_n = k|\pi) = \pi_k$: $p(y_n|\mathbf{x}_n, \beta, \pi) = \sum_k p(y_n, r_n = k|\mathbf{x}_n, \beta, \pi) = \sum_k p(y_n|r_n = k, \mathbf{x}_n, \beta, \pi)\pi_k = \sum_k \mathcal{N}(y_n|\beta_k^T \tilde{\mathbf{x}}_n, \sigma^2)\pi_k$. $-log p(\mathbf{y}|X, \beta, \pi) = -\sum_n log \sum_k \mathcal{N}(y_n|\beta_k^T \tilde{\mathbf{x}}_n, \sigma^2)\pi_k$. Model is not convex as a sum of gaussian. Not identifiable by permutation of labels.