EECS 3405 Final Review Formulas

Linear Regression: Model	$y = w^{\mathrm{T}}x$
Linear Regression: Loss Function	$E(w) = \frac{1}{2}\sum_{i=1}^{N}(w^{\mathrm{T}}x_{\mathrm{i}} - y_{\mathrm{i}})^2$
Linear Regression: Derivative	$\frac{\partial E}{\partial w} = X^{\mathrm{T}}Xw - X^{\mathrm{T}}y$
Linear Regression: Gradient for a particular point (x,y)	$\frac{\partial E}{\partial w}\frac{1}{2}(w^{\mathrm{T}}x - y)^2 = \frac{1}{2}\cdot 2(w^{\mathrm{T}}x - y)\frac{\partial E}{\partial w}(w^{\mathrm{T}}x - y)$ $= (w^{\mathrm{T}}x - y)x$ $\nabla E(w) = (x^{\top}w - y)x$
Linear Regression: Closed form of derivative by setting to 0	$w^* = (X^{\top}X)^{-1}X^{\top}y$

Stochastic Gradient Descent Method (SGD)	randomly choose $\mathbf{x}^{(0)}$, and set η_0
	 set $n = 0$
	 while not converged do
	 update: $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \eta_n \nabla f(\mathbf{x}^{(n)})$
	 adjust: $\eta_n \rightarrow \eta_{n+1}$
	 $n = n + 1$
	 end while

LDA	Supervised: $\hat{\mathrm{w}} = \arg\max_{\mathrm{w}} \frac{\mathbf{w}^{\top}\mathbf{S}_{\mathbf{b}}\mathbf{w}}{\mathbf{w}^{\top}\mathbf{S}_{\mathbf{w}}\mathbf{w}}$ $S_w^{-1}S_b w = \lambda w$

PCA	Unsupervised: $$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(v_i - \hat{v})(v_i - \hat{v})$$ $$\sigma^2 = w^{\top} S w$$
PCA Procedure	1 compute the sample covariance matrix S from training data (*measures how features vary)* 2 calculate the top m eigenvectors of S (*Eigenvectors corresponding to largest eigenvalues are the directions where the data has the most variance, Choose the top* ***m*** *of these to reduce the data to m dimensions.* ***Top*** the largest eigen value) Then, put those selected eigen vectors as a row of a matrix ("A"), size mxn: 3 form A ∈Rm×n with an eigenvector in a row $$\mathbf{A} = \begin{bmatrix} — & \hat{\mathbf{w}}_1^{\top} & — \\ — & \hat{\mathbf{w}}_2^{\top} & — \\ & \vdots & \\ — & \hat{\mathbf{w}}_m^{\top} & — \end{bmatrix}_{m\times n}$$ 4 for any x ∈Rn, map it to y ∈Rm as y= Ax.
Lagrange Theorem	$$\nabla f(x,y) = \lambda \nabla g(x,y)$$ Set $\nabla_x L = 0$ and $\nabla_y L = 0$ Example: f(x, y) = x y, Subject to: g(x, y) = x + y - 10 = 0 $$L(x,y,\lambda) = x\,y - \lambda(x + y - 10)$$ $$= x\,y - \lambda\,x - \lambda\,y + \lambda\,10$$ $$\frac{\partial L}{\partial x} = y - \lambda = 0 \quad ; \quad \frac{\partial L}{\partial y} = x - \lambda = 0 \quad ; \quad \frac{\partial L}{\partial \lambda} = x + y - 10 = 0$$ Solving this λ = x = y and x,y = 5 thus: f(5,5) = 25

MCE: Heavyside function which sums the errors	$$\mathrm{E_0(w)} = \sum_{i=1}^{N} \mathrm{H(\text{-}y_i w^T x_i)}$$
Sigmoid approximation of MCE	$$\mathrm{E_1(w)} = \sum_{i=1}^{N} \sigma(\text{-}\mathrm{y_i w^T x_i})$$
Gradient of Sigmoid Approximation:	Using derivative of Sigmoid: $$\frac{d}{dx}\sigma(x) = \sigma(x)\big(1-\sigma(x)\big)$$ $$\frac{\partial}{\partial w} l(-y_i w^T x_i) = l(-y_i w^T x_i)\big(1 - l(-y_i w^T x_i)\big) \cdot \frac{\partial}{\partial w}(-y_i w^\top x_i)$$ $$= -y_i l(-y_i w^T x_i)\big(1 - l(-y_i w^T x_i)\big)(x_i)$$ Thus: $$\frac{\partial E_1(w)}{\partial w} = -\sum_{i=1}^{N} y_i l(-y_i w^T x_i)\big(1 - l(-y_i w^T x_i)\big)(x_i)$$
2 ways to represent gradient of MCE:	$$\frac{\partial E_1(w)}{\partial w} = -\sum_{i=1}^{N} y_i l(-y_i w^T x_i)\big(1 - l(-y_i w^T x_i)\big)(x_i)$$ Or, with a simpler gradient: $$\frac{\partial E_2(w)}{\partial w} = \sum_{i=1}^{N} y_i l(-y_i w'^T x_i)\big(1 - l(y_i w'^T x_i)\big)(x_i)$$
The average MCE loss function, show also the vectorized form	Both are average loss (times 1/n) over all samples: $$\frac{1}{n}\sum_{i=1}^{N} l(-y_i w^T x_i)$$ $$\frac{1}{n}\sigma\big(-y \odot (Xw)\big)$$

Logistic Regression	$$w_{LR} = arg\ \max_{w} L(w) = arg\ \max_{w} \prod_{i}^{N} \sigma(y_i w^T x_i)$$ Where $\sigma = 1$
Logistic Regression in natural logarithmic form and how it is derived:	$$\ln\big(L(w)\big) = \ln\left(\prod_{i}^{N} \sigma(y_i w^T x_i)\right)$$ Log property: ln(ab)=lna + lnb $$ln(a \cdot b \cdot c \ldots.) = lna + lnb + lnc + \cdots$$ $$\ln\big(L(w)\big) = \sum_{i=1}^{N} \ln\big(\sigma(y_i w^T x_i)\big)$$
Derivative of Logistic Regression in natural log form:	Taking the gradient wrt w: $$\frac{\partial \ln\big(L(w)\big)}{\partial \mathrm{w}} = \sum_{i=1}^{N} \frac{\partial}{\partial \mathrm{w}} \ln\big(\sigma(y_i w^T x_i)\big)$$ $$= \sum_{i=1}^{N} \frac{1}{\sigma(y_i w^T x_i)} \cdot \frac{\partial}{\partial \mathrm{w}} \sigma(y_i w^T x_i)$$ Derivative of sigmoid function (plus chain rule): $$= \sum_{i=1}^{N} \frac{1}{\sigma(y_i w^T x_i)} \cdot \sigma(y_i w^T x_i)\big(1 - \sigma(y_i w^T x_i)\big)\,(y_i x_i)$$ $$= \sum_{i=1}^{N} \frac{1}{1} \cdot \big(1 - \sigma(y_i w^T x_i)\big)\,(y_i x_i)$$ $$= \sum_{i=1}^{N} \big(1 - \sigma(y_i w^T x_i)\big)\,(y_i x_i)$$

	Note: y_i is a scalar, thus acts like a coefficient: $$\frac{\partial \ln(L(w))}{\partial \mathrm{w}} = \sum_{i=1}^{N} \mathrm{y_i}(1-\sigma(y_i w^T x_i))x_i$$
How to represent the gradient for a particular mini batch of Logistic Regression (also with element-wise multiplication)	$$\frac{\partial Q(w;B)}{\partial B} = \frac{1}{B}\sum_{i=1}^{B} \mathrm{y_i}(1-\sigma(y_i w^T x_i))x_i$$ $$=\frac{1}{B}X^{\top}\left[y \odot l\left(y \odot (Xw)\right) - y\right]$$
Logistic Regression Loss Function	$$-\ln\left(\sigma\left(\boldsymbol{y} \odot (\boldsymbol{X}\boldsymbol{w})\right)\right) \qquad (\in \mathbb{R}^N)$$
SVM0 Formulation:	**max**$_{\varepsilon,w,b}$ γ $$\frac{y_i(w^T x_i\ +\ b)}{\lvert\lvert w \rvert\rvert} \geq\ \gamma, \forall_i \in\ \{1,2,\ldots,N\}$$
SVM1: Primal Problem	$$\text{Primal Problem:} \quad \min_{w,b}\frac{1}{2}w^T w \quad \text{subject to} \quad y_i(w^T x_i + b) \geq 1, \quad \forall i \in \{1,2,\ldots,N\}$$
How to represent SVM Primal Problem as a Lagrangian	$$\mathcal{L}(w,b,\alpha) = \frac{1}{2}\ w^T w - \sum_{i=1}^{N} \alpha_i(y_i(w^T x - b) - 1)$$ $$\mathcal{L}(w,b,\alpha) = \frac{1}{2}w^T w - \sum_{i=1}^{N} \alpha_i y_i w^T x_i + \alpha_i y_i b - \alpha_i$$

	$$\mathcal{L}(w,b,\alpha)=\frac{1}{2}w^Tw-\sum_{i=1}^{N}\alpha_i y_i w^T x_i - b\sum_{i=1}^{N}\alpha_i y_i+\sum_{i=1}^{N}\alpha_i$$
SVM2: Dual Problem	$$\max_{\alpha_i\geq 0}\left[\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j x_i^\top x_j\right]$$
Quadratic Programming Formulation	$$\max_{\boldsymbol{\alpha}} 1^\top\alpha-\frac{1}{2}\alpha^\top Q\alpha$$ **subject to:** $$y^T\alpha = 0$$ $$\alpha \geq 0$$
Define each matrix for Quadratic Programming:	$$\boldsymbol{\alpha}=\begin{bmatrix}\alpha_1\\ \vdots\\ \alpha_N\end{bmatrix}_{N\times 1}\quad \mathbf{y}=\begin{bmatrix}y_1\\ \vdots\\ y_N\end{bmatrix}_{N\times 1}\quad \mathbf{1}=\begin{bmatrix}1\\ \vdots\\ 1\end{bmatrix}_{N\times 1}$$ $$\mathbf{Q}=\begin{bmatrix}Q_{ij}\end{bmatrix}_{N\times N}=\begin{bmatrix}\mathbf{y}\mathbf{y}^\top\end{bmatrix}_{N\times N}\odot\begin{bmatrix}\mathbf{x}_1^\top\mathbf{x}_1 & \cdots & \mathbf{x}_1^\top\mathbf{x}_N\\ \vdots & \mathbf{x}_i^\top\mathbf{x}_j & \vdots\\ \mathbf{x}_N^\top\mathbf{x}_1 & \cdots & \mathbf{x}_N^\top\mathbf{x}_N\end{bmatrix}_{N\times N}$$ where $\odot$ denotes element-wise multiplication, and: $$Q_{ij} = (y_i y_j)\odot(x_i^T x_j)$$

Soft SVM Problem	$$\min_{w,b,\xi_i} \left(\frac{1}{2}\right) w^T w + C \sum_{i=1}^{N} \xi_i$$ **subject to:** $$y_i(w^T x_i + b) \geq 1 - \xi_i, \forall i \in \{1, 2, \ldots, N\}$$ $\xi_i \geq 0, \forall i \in \{1, 2, \ldots, N\}$
Soft SVM: Dual Problem + How is it different?	$$\max_{\alpha} 1^\top \alpha - \frac{1}{2} \alpha^\top Q \alpha$$ **subject to:** $$y^T \alpha = 0$$ $$0 \leq \alpha \leq C$$ the α's must be greater than 0, but less than C. That C is the hyperparameter for the maximum allowance of error.
L_2 norm	**L_2 norm:** $\|\|w\|\|_2 = \sqrt{(\|w_1\|^2 + \ldots + \|w_n\|^2)}$
Ridge Regression Objective Function **(also in elementwise multiplication)**	$$w^*_{ridge} = arg\min_{w} \left[\sum_{i=1}^{N} (w^T x_i - y_i)^2 + \lambda \cdot \|\|w\|\|_2^2 \right]$$ $$w^* = \arg\min_{w} \left[\sum_{i=1}^{N} \left(\sum_{j=1}^{d} w_j x_{ij} - y_i \right)^2 + \lambda \sum_{j=1}^{d} (w_j \odot w_j) \right]$$
How to derive the Ridge Regression Closed form Solution	$$\sum_{i=1}^{N} (w^T x_i - y_i)^2 + \lambda \cdot \|\|w\|\|_2^2$$ $$L_{ridge}(w) = \|\|Xw - y\|\|^2 + \lambda w^\top w$$ $$= (Xw - y)^\top (Xw - y) + \lambda w^\top w$$ $$= (Xw - y)^\top (Xw - y) + \lambda w^\top w$$ $$= w^\top X^\top X w - 2Xwy^\top + y^\top y + \lambda w^\top w$$ $$\nabla L_{ridge}(w) = 2X^\top X w - 2X^\top y + 2\lambda w$$

	Set gradient to 0: $$2X^{\top}Xw - 2X^{\top}y + 2\lambda w = 0$$ $$X^{\top}Xw - X^{\top}y + \lambda w = 0$$ $$(X^{\top}X + \lambda \mathrm{I})w = X^{\top}y$$ $$w^{*}_{ridge} = (\mathrm{X}^{\mathrm{T}}\mathrm{X} + \lambda\mathrm{I})^{-1}\mathrm{X}^{\mathrm{T}}\mathrm{y}$$
LASSO Objective Function	$$w^{*}_{lasso} = arg\min_{w}\left[\frac{1}{2}\sum_{i=1}^{N}(w^{T}x_i - y_i)^2 + \lambda \cdot \left\|\|w\|\right\|_1\right]$$
How to derive LASSO Gradient (component wise)	$$\mathrm{sign}(w_j) = \begin{cases} +1 & \text{if } w_j > 0 \\ -1 & \text{if } w_j < 0 \\ 0 & \text{if } w_j = 0 \text{ (convention for gradient descent)} \end{cases}$$ $$Q_{lasso}(w) = \frac{1}{2}\sum_{i=1}^{N}(w^{T}x_i - y_i)^2 + \lambda \cdot \left\|\|w\|\right\|_1$$ $$= \sum_{i=1}^{N}(w^{T}x_i - y_i)\cdot x_i + \lambda \cdot sign(w)$$ $$= \sum_{i=1}^{N}x_i(w^{T}x_i) - \sum_{i=1}^{N}y_i x_i \quad {}_{i} \cdot + \lambda \cdot sign(w)$$ $$= \sum_{i=1}^{N}x_i(x_i^{T}w) - \sum_{i=1}^{N}y_i x_i \cdot + \lambda \cdot sign(w)$$ $$= \left(\sum_{i=1}^{N}x_i x_i^{T}\right)w - \sum_{i=1}^{N}y_i x_i + \lambda \cdot sign(w)$$

Loss function for matrix factorization	$$Q(U,V) = \sum_{(i,j)\in\Omega} \left(x_{ij} - u_i^T v_j\right)^2 + \lambda_1 \sum_{i=1}^{n} \lvert u_i \rvert_2^2 + \lambda_2 \sum_{j=1}^{m} \lvert v_j \rvert_2^2$$
Neural Network Softmax function	$$y_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$
Forward Pass of a Fully-Connected DNN	1. For the input layer: $z_0 = x$ 2. For each hidden layer l = 1, 2, …, L-1: *The output from previous layer (al) equals the non-linear activation plus bias:* $$a_l = W^{(l)} z_{l-1} + b^l$$ $$z_l = ReLU(a_l)$$ 3. For the output layer: $$a_L = W^{(L)} z_{L-1} + b^{(L)}$$ $$y = z_L = softmax(a_L)$$
Derivatives for different automatic differentiation (full connection, nonlinear activation)	
Why is $x^T Qx$ considered quadratic? (where x is a column vector and Q is a matrix)	$$x^T Qx = \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{ij}\, , x_i x_j$$ **Every term is of degree 2**

How to solve a max / min problem based on a matrix Q:	• Solve $\nabla f = 0 \rightarrow$ this gives critical point • Check Hessian = 2Q • If $Q \succ 0 \rightarrow$ it's a **minimum** • If $Q \prec 0 \rightarrow$ it's a **maximum** If Q indefinite → **saddle**