Two If $D>N$ the task is understermined (more dimensions than data) regularization. Cost functions SE = $\frac{1}{N}\sum_{n=1}^{N}[y_n-f(x_n)]^2$ Not good it houtliers. MAE = $\frac{1}{N}\sum_{n=1}^{N} y_n-f(x_n) $ In Convexity In joining two points never intercts with the function anywhere else. $\lambda u + (1-\lambda)v) \le \lambda f(u) + (1-\lambda)f(v)$ with $\in [0;1]$. A strictly convex function has unique global minimum w^* . Sums of invex functions are convex. function must always lie above its ligarisation $\mathcal{L}(u) \ge \mathcal{L}(w) + \nabla \mathcal{L}(w)^T(u-v)$ set is convex iff the line segment bet- (XXT) y and $\hat{y}_m = x_m^T w^*$ Gram matrix invertible iff $rank(X) = D$ (use SVD if this is the case to get pseudo-inverse). Likelihood Probability of observing the data given a set of parameters and inputs: $p(y X,w) = \prod p(y_n x_n,w) = \prod \mathcal{N}(y_n x_n^Tw,\sigma^2)$ Best model maximises \log -likelihood $\mathcal{L}_{LL} = -\frac{1}{2\sigma^2}\sum (y_n - x_n^Tw)^2 + cst$. 6 Regularization 6.1 Ridge Regression $\mathcal{L}(w) = \frac{1}{2}(y - Xw)^T(y - Xw) + \frac{\lambda}{2} w _2^2 \rightarrow w^*_{ridge} = (XX^T + \lambda I_D)^{-1}X^Ty = X^T(XX^T + \lambda I_N)^{-1}y$ Can be considered a MAP estimator:	ultiple $y_n \approx f(\mathbf{x_n}) := w_0 + \sum_{j=1}^D w_j x_{nj} = 0$	$X^{\perp}(y - Xw) = 0 \Rightarrow w' =$
retermined (more dimensions than data) regularization. Cost functions $SE = \frac{1}{N} \sum_{n=1}^{N} y_n - f(x_n) ^2 \text{ Not good ith outliers. MAE} = \frac{1}{N} \sum_{n=1}^{N} y_n - f(x_n) $ It convexity line joining two points never intercts with the function anywhere else. $\lambda u + (1 - \lambda)v \geq \lambda f(u) + (1 - \lambda)f(v)$ with $\in [0;1]$. A strictly convex function has unique global minimum w^* . Sums of wax functions are convex. function must always lie above its liverarisation $\mathcal{L}(u) \geq \mathcal{L}(u) + \nabla \mathcal{L}(w)^T(u)$ arisation $\mathcal{L}(u) \geq \mathcal{L}(w) + \nabla \mathcal{L}(w)^T(u)$ are set is convex iff the line segment between any two points of \mathcal{C} lies in \mathcal{C} : $u + (1 - \theta)v \in \mathcal{C}$ Optimisation 1a Gradient $\nabla \mathcal{L} := \begin{bmatrix} \frac{\partial \mathcal{L}(w)}{\partial w_1} & \dots & \frac{\partial \mathcal{L}(w)}{\partial w_D} \end{bmatrix}$ 1a Gradient descent $(t+1) = w(t) - \gamma \nabla \mathcal{L}(w(t))$. Very sensitive ill-conditioning. D - Linear Reg D - Linear Reg D - Linear Reg S - $\frac{1}{N} \sum \mathcal{L}_n(w)$ with update $w(t+1) = w(t) - \gamma \nabla \mathcal{L}_n(w(t))$ and $weights = 2N * D + D$. 22 SGO $\frac{1}{N} \sum \mathcal{L}_n(w)$ with update $w(t+1) = w(t) - \gamma \nabla \mathcal{L}_n(w(t))$ 3 Mini-batch SGD $\frac{1}{N} \sum_{n \geq 0} \sum_{n \geq$,	$(XX^T)^{-1}X^Ty$ and $\hat{y}_m = x_m^Tw^*$ Gram
Tregularization. Cost functions $SE = \frac{1}{N}\sum_{n=1}^{N}[y_n - f(x_n)]^2$ Not good it houtliers. $MAE = \frac{1}{N}\sum_{n=1}^{N} y_n - f(x_n) ^2$ Not good it houtliers. $MAE = \frac{1}{N}\sum_{n=1}^{N} y_n - f(x_n) ^2$ Not good it houtliers. $MAE = \frac{1}{N}\sum_{n=1}^{N} y_n - f(x_n) ^2$ Not good it houtliers. $MAE = \frac{1}{N}\sum_{n=1}^{N} y_n - f(x_n) ^2$ Not good it houtliers. $MAE = \frac{1}{N}\sum_{n=1}^{N} y_n - f(x_n) ^2$ Not good it houtliers. $MAE = \frac{1}{N}\sum_{n=1}^{N} y_n - f(x_n) ^2$ Not good it houtliers. $MAE = \frac{1}{N}\sum_{n=1}^{N} y_n - f(x_n) ^2$ Not good it houtliers. $MAE = \frac{1}{N}\sum_{n=1}^{N} y_n - f(x_n) ^2$ Not good it houtliers. $MAE = \frac{1}{N}\sum_{n=1}^{N} y_n - f(x_n) ^2$ Not good it houtliers. $MAE = \frac{1}{N}\sum_{n=1}^{N} y_n - f(x_n) ^2$ Not good it houtliers. $MAE = \frac{1}{N}\sum_{n=1}^{N} y_n - f(x_n) ^2$ Not good it houtliers. $MAE = \frac{1}{N}\sum_{n=1}^{N} y_n - f(x_n) ^2$ Not good it houtliers $MEE = \frac{1}{N}\sum_{n=1}^{N} y_n - f(x_n) ^2$ Not good it houtliers $MEE = \frac{1}{N}\sum_{n=1}^{N} y_n - f(x_n) ^2$ Not good it houtliers $MEE = \frac{1}{N}\sum_{n=1}^{N} y_n - f(x_n) ^2$ Not good it houtliers $MEE = \frac{1}{N}\sum_{n=1}^{N} y_n - f(x_n) ^2$ Not good it houtliers $MEE = \frac{1}{N}\sum_{n=1}^{N} y_n - f(x_n) ^2$ Not MEE		
Cost functions $SE = \frac{1}{N} \sum_{n=1}^{N} y_n - f(x_n) ^2$ Not good ith outliers $AE = \frac{1}{N} \sum_{n=1}^{N} y_n - f(x_n) ^2$ Rot good ith outliers $AE = \frac{1}{N} \sum_{n=1}^{N} y_n - f(x_n) ^2$ line joining two points never interies with the function anywhere else. $Au + (1 - \lambda)v) \le \lambda f(u) + (1 - \lambda)f(v)$ with $\in [0,1]$. A strictly convex function has unique global minimum w^* . Sums of one functions are convex. functions are convex. function must always lie above its liar arisation $\mathcal{L}(u) \ge \mathcal{L}(w) + \nabla \mathcal{L}(w)^T (u - v)^T (u) \ge \mathcal{L}(w) + \nabla \mathcal{L}(w)^T (u - v)^T (u) \ge \mathcal{L}(w) + \nabla \mathcal{L}(w)^T (u - v)^T (u) \ge \mathcal{L}(w) + \nabla \mathcal{L}(w)^T (u - v)^T (u) \ge \mathcal{L}(w) + \nabla \mathcal{L}(w)^T (u - v)^T (u) \ge \mathcal{L}(w) + \nabla \mathcal{L}(w)^T (u - v)^T (u) \ge \mathcal{L}(w) + \nabla \mathcal{L}(w)^T (u - v)^T (u) \ge \mathcal{L}(w) + \frac{1}{2} w ^2 \ge \mathcal{L}(w)$	regularization.	
Probability of observing the data given a set of parameters and inputs: $p(y X,w) = 1$ Convexity line joining two points never intercts with the function anywhere else. $\lambda u + (1-\lambda) y \ge \lambda f(u) + (1-\lambda) f(y)$ with $\in [0;1]$. A strictly convex function has unique global minimum w^* . Sums of invex functions are convex. function must always lie above its licarisation $\mathcal{L}(u) \ge \mathcal{L}(w) + \nabla \mathcal{L}(w)^T (u - v) + (u - v) e \mathcal{L}(w)$ where any two points of \mathcal{C} lies in \mathcal{C} : $u + (1-\partial)v \in \mathcal{C}$ Optimisation Tardient $\nabla \mathcal{L} := \left[\frac{\partial \mathcal{L}(w)}{\partial w_1} \dots \frac{\partial \mathcal{L}(w)}{\partial w_D}\right]$ Tardient descent 11 Gradient descent 12 SGD $u = \frac{1}{N} \sum \mathcal{L}_n(w)$ with update $w^{(t+1)} = \frac{1}{N} \sum \mathcal{L}_n(w)$ with update $w^{(t+1)} = \frac{1}{N} \sum \mathcal{L}_n(w)$ with update $w^{(t+1)} = w^{(t)} - \gamma \nabla \mathcal{L}(w^{(t)})$. Sample subgradient at w cege the gradient: $\nabla \mathcal{L}_{MAE} = \frac{1}{N} \sum_n sgn(x_n) \nabla f(x_n)$. Sample subgradient for MAE: $h(e) = \mathcal{L}_{M} = \mathcal{L}_{N}(v) + \mathcal{L}_{M}(w)$. Sample subgradient for MAE: $h(e) = \mathcal{L}_{M} = \mathcal{L}_{M}(v) + \mathcal{L}_{M}(w)$. Sample subgradient for MAE: $h(e) = \mathcal{L}_{M} = \mathcal{L}_{M}(v) + \mathcal{L}_{M}(v)$. Sample subgradient for MAE: $h(e) = \mathcal{L}_{M} = \mathcal{L}_{M}(v) + \mathcal{L}_{M}(v)$. Sample subgradient for MAE: $h(e) = \mathcal{L}_{M} = \mathcal{L}_{M}(v) + \mathcal{L}_{M}(v)$. Sample subgradient for MAE: $h(e) = \mathcal{L}_{M}(v) + \mathcal{L}_{M}(v)$. Sample subgradient is $\nabla \mathcal{L}_{MAE} = \mathcal{L}_{M}(v) + \mathcal{L}_{M}(v)$. Sample subgradient for MAE: $h(e) = \mathcal{L}_{M}(v) + \mathcal{L}_{M}(v)$. Sample subgradient for $\mathcal{L}_{M}(v) + \mathcal{L}_$	Cost functions	'
set of parameters and inputs : $p(y X,w) = 1$ Convexity set of parameters and inputs : $p(y X,w) = 1$ $p(x X,w) = 1$	$SE = \frac{1}{N} \sum_{n=1}^{N} [y_n - f(x_n)]^2$ Not good	
In Convexity line joining two points never intersets with the function anywhere else. $\lambda u + (1-\lambda)v \ge \lambda f(u) + (1-\lambda)f(v)$ with $\in [0;1]$. A strictly convex function has unique global minimum w^* . Sums of onex functions are convex. function must always lie above its liariarisation $\mathcal{L}(u) \ge \mathcal{L}(w) + \nabla \mathcal{L}(w)^T(u-1)^T(u) + (1-\alpha)v \in \mathcal{L}(w)$. Set is convex iff the line segment betaen any two points of \mathcal{C} lies in \mathcal{C} : $u + (1-\theta)v \in \mathcal{C}$ Optimisation radient $\nabla \mathcal{L} := \left[\frac{\partial \mathcal{L}(w)}{\partial u} + \nabla \mathcal{L}(w)^T(v-1) + (1-\theta)v \in \mathcal{C}\right]$ Optimisation radient $\nabla \mathcal{L} := \left[\frac{\partial \mathcal{L}(w)}{\partial u} + \nabla \mathcal{L}(w)^T(v-1) + (1-\theta)v \in \mathcal{C}\right]$ Optimisation radient $\nabla \mathcal{L} := \left[\frac{\partial \mathcal{L}(w)}{\partial u} + (1-\theta)v \in \mathcal{C}\right]$ Optimisation radient $\nabla \mathcal{L} := \left[\frac{\partial \mathcal{L}(w)}{\partial u} + (1-\theta)v \in \mathcal{C}\right]$ Optimisation radient $\nabla \mathcal{L} := \left[\frac{\partial \mathcal{L}(w)}{\partial u} + (1-\theta)v \in \mathcal{C}\right]$ Optimisation radient $\nabla \mathcal{L} := \left[\frac{\partial \mathcal{L}(w)}{\partial u} + (1-\theta)v \in \mathcal{C}\right]$ Optimisation radient $\nabla \mathcal{L} := \left[\frac{\partial \mathcal{L}(w)}{\partial u} + (1-\theta)v \in \mathcal{C}\right]$ Optimisation radient $\nabla \mathcal{L} := \left[\frac{\partial \mathcal{L}(w)}{\partial u} + (1-\theta)v \in \mathcal{C}\right]$ Optimisation radient $\nabla \mathcal{L} := \left[\frac{\partial \mathcal{L}(w)}{\partial u} + (1-\theta)v \in \mathcal{C}\right]$ Optimisation radient $\nabla \mathcal{L} := \left[\frac{\partial \mathcal{L}(w)}{\partial u} + (1-\theta)v \in \mathcal{C}\right]$ Optimisation radient $\nabla \mathcal{L} := \left[\frac{\partial \mathcal{L}(w)}{\partial u} + (1-\theta)v \in \mathcal{C}\right]$ Optimisation radient $\nabla \mathcal{L} := \left[\frac{\partial \mathcal{L}(w)}{\partial u} + (1-\theta)v \in \mathcal{C}\right]$ Optimisation radient $\nabla \mathcal{L} := \left[\frac{\partial \mathcal{L}(w)}{\partial u} + (1-\theta)v \in \mathcal{C}\right]$ Optimisation radient $\nabla \mathcal{L} := \left[\frac{\partial \mathcal{L}(w)}{\partial u} + (1-\theta)v \in \mathcal{C}\right]$ Optimisation radient $\nabla \mathcal{L} := \left[\frac{\partial \mathcal{L}(w)}{\partial u} + (1-\theta)v \in \mathcal{L}(w)\right]$ 1 Gradient descent $(t+1) = w(t) - \gamma \nabla \mathcal{L}(w(t))$. Very sensitive illiconditioning. D. Linear Reg $(w) = \frac{1}{N} \nabla x^T (y - Xw)$. So $(w) = \frac{1}{N} \nabla x^T (y - Xw)$. So $(w) = \frac{1}{N} \nabla x^T (y - Xw)$. So $(w) = \frac{1}{N} \nabla x^T (y - Xw)$. So $(w) = \frac{1}{N} \nabla x^T (y - Xw)$. So $(w) = \frac{1}{N} \nabla x^T (y - Xw)$. So $(w) = \frac{1}{N} \nabla x^T (y - Xw)$. So $(w) = \frac{1}{N} \nabla x^T (y - Xw)$. So $(w) = \frac{1}{N} \nabla x^T (y - Xw)$. So $(w) = \frac{1}{N} \nabla x^T (y - Xw)$. So $(w) = \frac{1}{N} \nabla x^T ($	ith outliers. MAE = $\frac{1}{N} \sum_{n=1}^{N} y_n - f(x_n) $	
line joining two points never interes with the function anywhere else. $\lambda u + (1-\lambda)v \le Af(u) + (1-\lambda)f(v)$ with $\in [0;1]$. A strictly convex function has unique global minimum w^* . Sums of movex functions are convex. function must always lie above its liearisation $\mathcal{L}(u) \ge \mathcal{L}(w) + \nabla \mathcal{L}(w)^T(u)$ earisation $\mathcal{L}(u) \ge \mathcal{L}(w) + \nabla \mathcal{L}(w)^T(u)$ by u, w . Set is convex iff the line segment beter any two points of \mathcal{C} lies in \mathcal{C} : $u + (1-\theta)v \in \mathcal{C}$ Optimisation radient $\nabla \mathcal{L} := \left[\frac{\partial \mathcal{L}(w)}{\partial w_0} \dots \frac{\partial \mathcal{L}(w)}{\partial w_0}\right]$ 1 Gradient descent $(t+1) = w(t) - \gamma \nabla \mathcal{L}(w(t))$. Very sensitive ill-conditioning. D - Linear Reg $(w) = \frac{1}{2N} (y - Xw)^T (y - Xw)$ and $weights = 2N * D + D$. Sot $: O_{error}(N * D) = 2N * D + N$ and $weights = 2N * D + D$. 2 SGD $= \frac{1}{N} \sum \mathcal{L}_n(w)$ with update $w(t+1) = w(t) - \gamma \mathcal{D} \mathcal{L}_n(w(t))$. 3 Mini-batch SGD $= \frac{1}{ k } \sum_{n \in \mathbb{R}} \nabla \mathcal{L}_n(w(t))$ with update $(t+1) = w(t) - \gamma g$. 4 Subgradient at $w \in \mathbb{R}^D$ such that $\mathcal{L}(u) \ge \mathcal{L}(w) + \mathbf{g}^T(u - w)$. cample subgradient for MAE: $h(e) = \mathcal{L}_n = \frac{1}{N} \sum_n sgn(x_n) \nabla f(x_n)$. 5 Projected SGD $(t+1) = \mathcal{P}_C[w(t) - \gamma \nabla \mathcal{L}_w(t)]$ 6 Newton's method for cond order (more expensive $O(ND^2 + \frac{N}{3})$ but faster convergence).		
cts with the function anywhere else. $\lambda u + (1 - \lambda)v \le \lambda f(u) + (1 - \lambda)f(v)$ with $\in [0;1]$. A strictly convex function has unique global minimum w^* . Sums of movex functions are convex. function must always lie above its librarisation $\mathcal{L}(u) \ge \mathcal{L}(w) + \nabla \mathcal{L}(w)^T (u - v)^T (u - v)^T (u) = \lambda v^T (u$		Best model maximises log-likelihood
All $+(1-A D) = \lambda I(u) + (1-A D$	cts with the function anywhere else.	
unique global minimum w^* . Sums of onex function are convex. function must always lie above its livarisation $\mathcal{L}(u) \geq \mathcal{L}(w) + \nabla \mathcal{L}(w)^T (u - v)^T (u)$, we set is convex iff the line segment beteen any two points of \mathcal{C} lies in \mathcal{C} : $u + (1 - \theta) v \in \mathcal{C}$ Optimisation radient $\nabla \mathcal{L} := \left[\frac{\partial \mathcal{L}(w)}{\partial w_1} \dots \frac{\partial \mathcal{L}(w)}{\partial w_D}\right]$ 1 Gradient descent $(t+1) = w^{(t)} - \gamma \nabla \mathcal{L}(w^{(t)})$. Very sensitive ill-conditioning. D-Linear Reg $(w) = \frac{1}{2N}(v - Xw)^T (v - Xw)$ and $(w) = \frac{1}{2N}(v - Xw)^T (v - Xw)$. Sost $(Oerropr(N * D) = 2N * D + N)$ and $(w) = \frac{1}{N} \sum \mathcal{L}_n(w)$ with update $(v) = \frac{1}{N} \sum \mathcal{L}_n(w)$ with	$(\lambda u + (1 - \lambda)v) \le \lambda f(u) + (1 - \lambda)f(v)$ with	
functions are convex. function must always lie above its lifearisation $\mathcal{L}(u) \geq \mathcal{L}(w) + \nabla \mathcal{L}(w)^T(u)$ set is convex iff the line segment beteen any two points of \mathcal{C} lies in \mathcal{C} : $w^*_{ridge} = (XX^T + \lambda I_D)^{-1}X^Ty = X^T(XX^T + \lambda I_N)^{-1}y$ can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered and program is $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered and program is $w^*_{ridge} = argmin_w - log(p(w X,y))$ can be considered ander $w^*_{ridge} = argmi$	\in [0;1]. A strictly convex function has	
function must always lie above its livarisation $\mathcal{L}(u) \geq \mathcal{L}(w) + \nabla \mathcal{L}(w)^T(u - v)^T(u)$ set is convex iff the line segment better any two points of \mathcal{C} lies in \mathcal{C} : $u + (1 - \theta)v \in \mathcal{C}$ Optimisation radient $\nabla \mathcal{L} := \left[\frac{\partial \mathcal{L}(w)}{\partial w_1} \dots \frac{\partial \mathcal{L}(w)}{\partial w_D}\right]$ 1 Gradient descent $(t+1) = w^{(t)} - \gamma \nabla \mathcal{L}(w^{(t)})$. Very sensitive ill-conditioning. D-Linear Reg $(w) = \frac{1}{2N}(y - Xw)^T(y - Xw) \rightarrow \mathcal{L}(w) = \frac{1}{2N}(y - Xw)^T(y - Xw)$ Sot: $O_{erropr}(N * D) = 2N * D + N$ and weights $= 2N * D + D$. 2 SGD $= \frac{1}{N} \sum \mathcal{L}_n(w)$ with update $w^{(t+1)} = \frac{1}{N} \sum \mathcal{L}_n(w)$ in $w \in \mathbb{R}^D$ such that $\mathcal{L}(u) \geq \mathcal{L}(w) + \mathbf{g}^T(u - w)$. cample subgradient for MAE: $h(e) = \frac{1}{N} \sum_n sgn(x_n) \nabla f(x_n)$. 5 Projected SGD 6 Newton's method cond order (more expensive $O(ND^2 + \frac{N}{N})$ but faster convergence). $w^*_{ridge} = argmin_w - log(p(w X, y))$ Can be considered a MAP estimator: $w^*_{ridge} = argmin_w - log(p(w X, y))$ $w^*_{ridge} = argmin_w - log(p(w $		
rarisation $\mathcal{L}(u) \geq \mathcal{L}(w) + \nabla \mathcal{L}(w)^T (u - w)^T ($		
Set is convex iff the line segment beteen any two points of \mathcal{C} lies in \mathcal{C} : $u+(1-\theta)v\in\mathcal{C}$ Optimisation radient $\nabla\mathcal{L}:=\left[\frac{\partial\mathcal{L}(w)}{\partial w_1} \dots \frac{\partial\mathcal{L}(w)}{\partial w_D}\right]$ 1 Gradient descent $(t+1)=w(t)-\gamma\nabla\mathcal{L}(w(t))$. Very sensitive ill-conditioning. D- Linear Reg $(w)=\frac{1}{2N}(y-Xw)^T(y-Xw)$ $\mathcal{L}(w)=-\frac{1}{N}X^T(y-Xw)$ Sost: $O_{error}(N*D)=2N*D+N$ and weights $=2N*D+D$. 2 SGD $=\frac{1}{N}\sum\mathcal{L}_n(w)$ with update $w^{(t+1)}=w^{(t)}-\gamma\nabla\mathcal{L}_n(w^{(t)})$. 3 Mini-batch SGD $=\frac{1}{ B }\sum_{n\in B}\nabla\mathcal{L}_n(w^{(t)})$ with update $w^{(t+1)}=w^{(t)}-\gamma g$. 4 Subgradient at $w\in\mathbb{R}^D$ such that $\mathcal{L}(u)\geq\mathcal{L}(w)+\mathbf{g}^T(u-w)$. Each $v\in\mathbb{R}^D$ such that $\mathcal{L}(u)\geq\mathcal{L}(w)+\mathbf{g}^T(u-w)$. Each $v\in\mathbb{R}^D$ such that $\mathcal{L}(u)\geq\mathcal{L}(w)+\mathbf{g}^T(u-w)$. So get the gradient: $\nabla\mathcal{L}_{MAE}=v$ $v\in\mathbb{R}^D$ such that $\mathcal{L}(u)\geq\mathcal{L}(w)+\mathbf{g}^T(u-w)$. So get the gradient: $\nabla\mathcal{L}_{MAE}=v$ $v\in\mathbb{R}^D$ such that $\mathcal{L}(u)\geq\mathcal{L}(w)+\mathbf{g}^T(u-w)$. So get the gradient: $\nabla\mathcal{L}_{MAE}=v$ $v\in\mathbb{R}^D$ such that $\mathcal{L}(u)\geq\mathcal{L}(w)+\mathbf{g}^T(u-w)$. So $v\in\mathbb{R}^D$ such that	•	8 .
set is convex iff the line segment betten any two points of \mathcal{C} lies in \mathcal{C} : $w_{ridge}^* = argmin_w - log(p(w X,y))$ $condomination$ radient $\nabla \mathcal{L} := \left[\frac{\partial \mathcal{L}(w)}{\partial w_1} \dots \frac{\partial \mathcal{L}(w)}{\partial w_D}\right]$ The Gradient descent of the line are segment better ill-conditioning. The Linear Reg (w) = $\frac{1}{2N}(y - Xw)^T(y - Xw)$ The set is $\frac{1}{2N}(y - Xw)^T(y - Xw)$		$(\lambda I_N)^{-1}y$
een any two points of \mathcal{C} lies in \mathcal{C} : $w_{ridge} = argmin_w - log(p(w X,y))$ 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. $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 7 Model Selection 7.1 Bias-Variance decomposition Small dimensions: large bias, small variance. Large dimensions: small bias, large variance. $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 8.2 Large dimensions: small bias, large variance. $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 8.2 Logistic regression $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 8.1 Optimal $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 8.2 Large dimensions: small bias, large variance. $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 8.2 Logistic regression $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 8.2 Logistic regression $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 8.2 Logistic regression $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 8.2 Logistic regression $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 8.1 Large dimensions: large bias, small variance. Large dimensions: small bias, large variance. $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 8.2 Logistic regression $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 8.2 Logistic regression $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 8.2 Logistic regression $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 8.2 Logistic regression $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 8.2 Logistic regression $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 8.2 Logistic regression $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 8.2 Logistic regression $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 8.2 Logistic regression $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 8.2 Logistic regression $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 8.2 Logistic regression $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 8.2 Logistic regression $w = \frac{1}{2N}(y - Xw)^T(y - Xw) + \lambda w _1$ 8.2 Logistic regr		
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. 8 Classification 8.1 Optimal $\hat{y}(x) = argmax_{y \in y} p(y x)$ 8.2 Logistic regression $weights = 2N * D + D.$ 2 SGD $= \frac{1}{N} \sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \frac{e^2}{N} \sum_{n \in \mathbb{N}} v_n(w^{(t)})$ 3 Mini-batch SGD $= \frac{1}{ B } \sum_{n \in \mathbb{N}} v_n(w^{(t)})$ 3 Mini-batch SGD $= \frac{1}{ B } \sum_{n \in \mathbb{N}} v_n(w^{(t)})$ 3 Wini-batch SGD $= \frac{1}{ B } \sum_{n \in \mathbb{N}} v_n(w^{(t)})$ 3 Wini-batch SGD $= \frac{1}{ B } \sum_{n \in \mathbb{N}} v_n(w^{(t)})$ 3 Wini-batch SGD $= \frac{1}{ B } \sum_{n \in \mathbb{N}} v_n(w^{(t)})$ 3 Wini-batch SGD $= \frac{1}{ B } \sum_{n \in \mathbb{N}} v_n(w^{(t)})$ 3 Wini-batch SGD $= \frac{1}{ B } \sum_{n \in \mathbb{N}} v_n(w^{(t)})$ 3 Wini-batch SGD $= \frac{1}{ B } \sum_{n \in \mathbb{N}} v_n(w^{(t)})$ 3 Wini-batch SGD $= \frac{1}{ B } \sum_{n \in \mathbb{N}} v_n(w^{(t)})$ 3 Wini-batch SGD $= \frac{1}{ B } \sum_{n \in \mathbb{N}} v_n(w^{(t)})$ 3 Wini-batch SGD $= \frac{1}{ B } \sum_{n \in \mathbb{N}} v_n(w^{(t)})$ 3 Wini-batch SGD $= \frac{1}{ B } \sum_{n \in \mathbb{N}} v_n(w^{(t)})$ 3 Wini-batch SGD $= \frac{1}{ B } \sum_{n \in \mathbb{N}} v_n(w^{(t)})$ 4 Subgradient at w $= \mathbb{R}^D \text{ such that } \mathcal{L}(u) \ge \mathcal{L}(w) + \mathbf{g}^T(u - w).$ Sample subgradient for MAE: $h(e) = v_n(e) = v_$	een any two points of $\mathcal C$ lies in $\mathcal C$:	$w_{ridge}^* = argmin_w - log(p(w X,y))$
radient $\nabla \mathcal{L} := \left[\frac{\partial \mathcal{L}(w)}{\partial w_1} \dots \frac{\partial \mathcal{L}(w)}{\partial w_D}\right]$ 1 Gradient descent (t+1) = $w^{(t)} - \gamma \nabla \mathcal{L}(w^{(t)})$. Very sensitive ill-conditioning. D - Linear Reg ($w) = \frac{1}{2N}(y - Xw)^T(y - Xw) \rightarrow \mathcal{L}(w) = -\frac{1}{N}X^T(y - Xw)$. Sost : $O_{error}(N*D) = 2N*D + N$ and weights = $2N*D + D$. 2 SGD = $\frac{1}{N} \sum \mathcal{L}_n(w)$ with update $w^{(t+1)} = \frac{e^z}{1}$ to limit the predicted values $w^{(t+1)} = \frac{e^z}{1}$ to $\frac{e^z}{1}$ ($\frac{e^z}{1}$ ($\frac{e^z}{1}$)		
1 Gradient descent $(t+1) = w^{(t)} - \gamma \nabla \mathcal{L}(w^{(t)})$. Very sensitive ill-conditioning. D - Linear Reg $(w) = \frac{1}{2N}(y - Xw)^T(y - Xw) \rightarrow \mathcal{L}(w) = -\frac{1}{N}X^T(y - Xw)$. Set: $O_{error}(N*D) = 2N*D + N$ and weights $= 2N*D + D$. 2 SGD $= \frac{1}{N}\sum \mathcal{L}_n(w)$ with update $w^{(t+1)} = \frac{e^2}{1+e^2}$ to limit the predicted values $y \in [0;1]$ $(p(1 x) = \sigma(x^Tw))$ and $p(0 x) = 1 - \sigma(x^Tw)$. Likelihood $p(y X,w) = \prod p(y_n x_n) = \prod n: y_n = 0 p(y_n = 0) x_n = 1 x_n = 0 x_n = 1 x_n = 1 x_n = 1 x_n = 0 x_n = 1 x_n = 1$		
1 Gradient descent $(t+1) = w^{(t)} - \gamma \nabla \mathcal{L}(w^{(t)})$. Very sensitive ill-conditioning. D - Linear Reg $(w) = \frac{1}{2N}(y - Xw)^T(y - Xw) \rightarrow \mathcal{L}(w) = -\frac{1}{N}X^T(y - Xw)$. Set: $O_{error}(N*D) = 2N*D + N$ and weights $= 2N*D + D$. 2 SGD $= \frac{1}{N}\sum \mathcal{L}_n(w)$ with update $w^{(t+1)} = \frac{e^2}{1+e^2}$ to limit the predicted values $y \in [0;1]$ $(p(1 x) = \sigma(x^Tw))$ and $p(0 x) = 1 - \sigma(x^Tw)$. Likelihood $p(y X,w) = \prod p(y_n x_n) = \prod n: y_n = 0 p(y_n = 0) x_n = 1 x_n = 0 x_n = 1 x_n = 1 x_n = 1 x_n = 0 x_n = 1 x_n = 1$	radient $\nabla \mathcal{L} := \begin{bmatrix} \frac{\partial \mathcal{L}(w)}{\partial x^2} & \dots & \frac{\partial \mathcal{L}(w)}{\partial x^2} \end{bmatrix}$	/
Small dimensions : large bias, small variance. Large dimensions : small bias, large variance. Small dimensions : large bias, small variance. Large dimensions : small bias, large variance. Small dimensions : large bias, small variance. Large dimensions : small bias, large variance. Small dimensions : large bias, small variance. Large dimensions : small bias, large variance. Small dimensions : large bias, small variance. Large dimensions : small bias, large variance. Small dimensions : large bias, small variance. Large dimensions : small bias, large variance. Small dimensions : large bias, small variance. Large dimensions : small bias, large variance. Small dimensions : large bias, small variance. Large dimensions : small bias, large variance. Small dimensions : large bias, small variance. Large dimensions : small bias, large variance. Small dimensions : large bias, small variance. Large dimensions : small bias, large variance. Small variance. Large dimensions : small bias, large variance. Small variance. Small variance. Large dimensions : small bias, large variance. Small variance. Large dimensions : small bias, large variance. Small variance specifical values $\hat{y}(x) = arg max_y \in yp(y x)$ State $\hat{y}(x) = p(y_1 x) = p(y_1 x) = p(y_2 x) = p(y_3 x) = p($		
ill-conditioning. D - Linear Reg $(w) = \frac{1}{2N}(y - Xw)^T(y - Xw) \rightarrow \mathcal{L}(w) = -\frac{1}{N}X^T(y - Xw)$. So $(x) = -\frac{1}{N}X^T(y $		
D- Linear Reg $(w) = \frac{1}{2N}(y - Xw)^T(y - Xw) \rightarrow \mathcal{L}(w) = \frac{1}{N}X^T(y - Xw).$ $\hat{y}(x) = argmax_{y \in y}p(y x)$ $\hat{y}(x) = argmax$		
$\mathcal{L}(w) = \frac{1}{2N}(y - Xw)^T(y - Xw) \rightarrow \\ \mathcal{L}(w) = -\frac{1}{N}X^T(y - Xw). \\ \text{ost} : O_{error}(N*D) = 2N*D+N \text{ and } \\ \text{weights} = 2N*D+D. \\ \textbf{2 SGD} \\ = \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = \\ \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{($		
$\mathcal{L}(w) = -\frac{1}{N}X^T(y - Xw).$ $\text{ost} : O_{error}(N * D) = 2N * D + N \text{ and}$ $\text{weights} = 2N * D + D.$ $\mathbf{Z} \text{ SGD}$ $= \frac{1}{N}\sum \mathcal{L}_n(w) \text{ with update } w^{(t+1)} = 0$ $\frac{1}{ B }\sum_{n\in B}\nabla \mathcal{L}_n(w^{(t)}).$ $\mathbf{Z} \text{ Subgradient at } w$ $\mathbf{Z} \text{ Subgradient at } w$ $\mathbf{Z} \text{ Subgradient for } MAE : h(e) = 0$ $\mathbf{Z} \text{ Subgradient } \mathbf{Z} \text{ such that } \mathcal{L}(u) \geq \mathcal{L}(w) + \mathbf{g}^T(u - w).$ $\mathbf{Z} \text{ sample subgradient } \mathbf{Z} \text{ such that } \mathcal{L}(u) \geq \mathcal{L}(w) + \mathbf{g}^T(u - w).$ $\mathbf{Z} \text{ subgradient } \mathbf{Z} \text{ such that } \mathcal{L}(u) \geq \mathcal{L}(w) + \mathbf{g}^T(u - w).$ $\mathbf{Z} \text{ subgradient } \mathbf{Z} \text{ such that } \mathcal{L}(u) \geq \mathcal{L}(w) + \mathbf{g}^T(u - w).$ $\mathbf{Z} \text{ subgradient } \mathbf{Z} \text{ such that } \mathcal{L}(u) \geq \mathcal{L}(w) + \mathbf{g}^T(u - w).$ $\mathbf{Z} \text{ subgradient } \mathbf{Z} \text{ such that } \mathcal{L}(u) \geq \mathcal{L}(w) + \mathbf{g}^T(u - w).$ $\mathbf{Z} \text{ subgradient } \mathbf{Z} \text{ such that } \mathcal{L}(u) \geq \mathcal{L}(w) + \mathbf{g}^T(u - w).$ $\mathbf{Z} \text{ subgradient } \mathbf{Z} \text{ such that } \mathcal{L}(u) \geq \mathcal{L}(w) + \mathbf{g}^T(u - w).$ $\mathbf{Z} \text{ subgradient } \mathbf{Z} \text{ such that } \mathcal{L}(u) \geq \mathcal{L}(w) + \mathbf{g}^T(u - w).$ $\mathbf{Z} \text{ subgradient } \mathbf{Z} \text{ such that } \mathcal{L}(u) \geq \mathcal{L}(w) + \mathbf{g}^T(u - w).$ $\mathbf{Z} \text{ subgradient } \mathbf{Z} \text$		
Solution $S_{weights} = 2N * D + D$. Solution $S_{weights} = 2N * D $		
weights = $2N*D+D$. 2 SGD $=\frac{1}{N}\sum \mathcal{L}_{n}(w) \text{ with update } w^{(t+1)} = \begin{cases} y \in [0;1] \ (p(1 x) = \sigma(x^{T}w) \text{ and } p(0 x) = 1 - \sigma(x^{T}w)). \\ \text{Likelihood} \\ p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0) \\ p(y X,w) = \prod_{n:y_{n}=K} p(y_{n} = 0) \\ p(y X,w) = \prod_{n:y_{n$		
2 SGD $= \frac{1}{N} \sum \mathcal{L}_{n}(w) \text{ with update } w^{(t+1)} = \begin{cases} y \in [0;1] \ (p(1 x) = \sigma(x^{T}w) \text{ and } p(0 x) = \\ 1 - \sigma(x^{T}w)). \\ \text{Likelihood} \end{cases}$ 3 Mini-batch SGD $= \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_{n}(w^{(t)}) \text{ with update } w^{(t+1)} = \begin{cases} y \in [0;1] \ (p(1 x) = \sigma(x^{T}w) \text{ and } p(0 x) = \\ 1 - \sigma(x^{T}w)). \\ \text{Likelihood} \end{cases}$ $= \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_{n}(w^{(t)}) \text{ with update } w^{(t+1)} = w^{(t)} - \gamma g.$ 4 Subgradient at w $\in \mathbb{R}^{D} \text{ such that } \mathcal{L}(u) \geq \mathcal{L}(w) + g^{T}(u - w).$ $\text{Example subgradient for MAE : } h(e) = w^{(t+1)} = y^{(t)} = y^{$	$Ost : O_{error}(N * D) = 2N * D + N $ and $Ost : O_{error}(N * D) = 2N * D + D.$	
$=\frac{1}{N}\sum \mathcal{L}_{n}(w) \text{ with update } w^{(t+1)} = y \in [0;1] \ (p(1 x) = \sigma(x^{T}w) \text{ and } p(0 x) = 1 - \sigma(x^{T}w)).$ Likelihood $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0) = 1 - \sigma(x^{T}w)).$ Likelihood $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0) = 1 - \sigma(x^{T}w)).$ Likelihood $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0) = 1 - \sigma(x^{T}w)).$ Likelihood $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0) = 1 - \sigma(x^{T}w)$ where $tildev_{n} = 1 \text{ if } y_{n} = k.$ For binary classification $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0) = 1 - \sigma(x^{T}w)$ where $tildev_{n} = 1 \text{ if } y_{n} = k.$ For binary classification $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0) = 1 - \sigma(x^{T}w)$ Sometime is eget the gradient $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0) = 1 - \sigma(x^{T}w)$ Loss $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0) = 1 - \sigma(x^{T}w)$ Where $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0)$ $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0)$ $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0)$ $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0)$ $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0)$ $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0)$ Likelihood $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0)$ $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0)$ $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0)$ $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0)$ $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0)$ $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0)$ $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0)$ $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0)$ $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0)$ $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0)$ $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0)$ $p(y X,w) = \prod p(y_{n} x_{n}) = \prod_{n:y_{n}=0} p(y_{n} = 0)$ $p(y X,w) = \prod p(y_{n} x_{n$		$\sigma(z) = \frac{e}{1+e^z}$ to limit the predicted values
$ \begin{array}{lll} & 1 - \sigma(x^T w)). \\ 3 & \mathbf{Mini-batch SGD} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{with update} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{where} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{where} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{where} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{where} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{where} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{where} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{where} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{where} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{where} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{where} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) & \text{where} \\ & = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t$		$y \in [0; 1] (p(1 x) = \sigma(x^T w) \text{ and } p(0 x) =$
3 Mini-batch SGD $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) \text{ with update} $ $ = \frac{1}{ B } \sum_{n \in B} \nabla \mathcal{L}_n(w^{(t)}) wit$		
$=\frac{1}{ B }\sum_{n\in B}\nabla\mathcal{L}_{n}(w^{(t)}) \text{ with update } 0 x_{n})\prod_{n:y_{n}=K}p(y_{n}=K x_{n})=\frac{1}{ B }\sum_{n\in B}\nabla\mathcal{L}_{n}(w^{(t)}) \text{ with update } 0 x_{n})\prod_{n:y_{n}=K}p(y_{n}=K x_{n})=\frac{1}{ B }\sum_{n\in B}\nabla\mathcal{L}_{n}(w^{(t)}) \text{ with update } 0 x_{n})\prod_{n:y_{n}=K}p(y_{n}=K x_{n})=\frac{1}{ B }\sum_{n\in B}\nabla\mathcal{L}_{n}(w^{(t)})=\frac{1}{ B }\sum_{n\in B}\nabla\mathcal{L}_{n}(w^{(t)$		
4 Subgradient at w $\in \mathbb{R}^D \text{ such that } \mathcal{L}(u) \geq \mathcal{L}(w) + \mathbf{g}^T(u - w).$ For binary classification $p(y X,w) = \prod p(y_n x_n) = \prod n:y_n=0 p(y_n = 1 \mid x_n) = 1 \mid x_n \mid x_n$		
4 Subgradient at w $tildey_{nk} = 1$ if $y_n = k$. For binary classification $p(y X,w) = \prod p(y_n x_n) = \prod_{n:y_n=0} p(y_n = w)$ ample subgradient for MAE: $h(e) = w \to g(e) = sgn(e)$ if $e \neq 0$, λ otherwise. If $e = get$ the gradient: $\nabla \mathcal{L}_{MAE} = \frac{1}{N} \sum_n sgn(x_n) \nabla f(x_n)$. So Projected SGD $\mathcal{L}_{MAE} = \mathcal{L}_{MAE} = \mathcal{L}_{MAE} = \mathcal{L}_{MAE} = \frac{1}{N} \sum_n sgn(x_n) \nabla f(x_n)$. So Projected SGD $\mathcal{L}_{MAE} = \mathcal{L}_{MAE} = \mathcal{L}_{MAE} = \frac{1}{N} \sum_n sgn(x_n) \nabla f(x_n)$. So $\mathcal{L}_{MAE} = \frac{1}{N} \sum_n sgn(x_n) \nabla f(x_n)$. So $\mathcal{L}_{MAE} = \frac{1}{N} \sum_n sgn(x_n) \nabla f(x_n)$. So $\mathcal{L}_{MAE} = \frac{1}{N} \sum_n sgn(x_n) \nabla f(x_n)$. So $\mathcal{L}_{MAE} = \frac{1}{N} \sum_{n=1}^{N} ln(1 + exp(x_n^T w)) - y_n x_n^T w$ which is convex in w . Gradient $\nabla \mathcal{L}_{MAE} = \frac{1}{N} \sum_{n=1}^{N} sgn(x_n) \nabla f(x_n) \nabla$, 11
For binary classification $p(y X,w) = \prod p(y_n x_n) = \prod_{n:y_n=0} p(y_n = x_n) = x_n + y_n + y_n = x_n + y_n + y_n$		
kample subgradient for MAE: $h(e) = p(y X, w) = \prod p(y_n x_n) = \prod_{n:y_n=0} p(y_n = x_n) = p(y_n$		
	$\in \mathbb{R}^D$ such that $\mathcal{L}(u) \ge \mathcal{L}(w) + \mathbf{g}^T (u - w)$.	For dinary classification $p(x Y x) = \prod_{x \in X} p(x x) = \prod_{x \in X} p(x x)$
e get the gradient : $\nabla \mathcal{L}_{MAE} = \prod_{n=1}^{N} \sigma(x_{n}^{T}w)^{y_{n}}[1 - \sigma(x_{n}^{T}w)]^{1-y_{n}}$ $\sum_{n} sgn(x_{n}) \nabla f(x_{n}).$ 5 Projected SGD $(t+1) = \mathcal{P}_{\mathcal{C}}[w^{(t)} - \gamma \nabla \mathcal{L}(w^{(t)})]$ 6 Newton's method $(t+1) = \mathcal{P}_{\mathcal{C}}[w^{(t)} - \gamma \nabla \mathcal{L}(w^{(t)})]$ $(t+1) = \mathcal{P}_{\mathcal{C}}[w^{(t)} - \gamma \nabla \mathcal{L}(w^{(t)})]$ $(t+1) = \mathcal{P}_{\mathcal{C}}[w^{(t)} - \gamma \nabla \mathcal{L}(w^{(t)})]$ which is convex in w . Gradient $\nabla \mathcal{L}(w) = \sum_{n=1}^{N} x_{n}(\sigma(x_{n}^{T}w) - y_{n}) = X^{T}[\sigma(Xw) - y] \text{ (no closed form solution)}.$		
$\frac{1}{N} \sum_{n} sgn(x_{n}) \nabla f(x_{n}).$ The interval of the project of the projec	$\rightarrow g(e) = sgn(e)$ if $e \neq 0$, λ otherwise.	
5 Projected SGD $\mathcal{L}(w) = \sum_{n=1}^{N} \ln(1 + exp(x_n^T w)) - y_n x_n^T w$ which is convex in w . 6 Newton's method $\text{cond order (more expensive } O(ND^2 + X^T [\sigma(Xw) - y] \text{ (no closed form solution).}$		$\prod_{n=0}^{N} \sigma(x_n^{1} w)^{y_n} [1 - \sigma(x_n^{1} w)]^{1-y_n}$
which is convex in w . Gradient $\nabla \mathcal{L}(w) = \sum_{n=1}^{N} x_n (\sigma(x_n^T w) - y_n) = X^T [\sigma(Xw) - y]$ (no closed form solution).		Loss $f(w) = \sum_{i=1}^{N} \ln(1 + \exp(x^T w)) = v_{i} x^T w$
Gradient $\nabla \mathcal{L}(w) = \mathcal{V}(w^T)$ Gradient $\nabla \mathcal{L}(w) = \sum_{n=1}^{N} x_n (\sigma(x_n^T w) - y_n) = X^T [\sigma(Xw) - y]$ (no closed form solution).		which is convex in w .
econd order (more expensive $O(ND^2 + X^T[\sigma(Xw) - y])$ (no closed form solution).	$\mathcal{P}_{\mathcal{C}}[w^{(t)} - \gamma \vee \mathcal{L}(w^{(t)})]$	Gradient
3) but faster convergence). $\begin{bmatrix} x & [b(xw) - y] \\ tion \end{bmatrix}$	o Newton's method	$\nabla \mathcal{L}(w) = \sum_{n=1}^{N} x_n (\sigma(x_n^T w) - y_n) =$
) but laster convergence). tion).		$X^{T}[\sigma(Xw) - y]$ (no closed form solu-
Hessian		tion).
	$W^{(i)} = W^{(i)} - \gamma^{(i)} (H^{(i)})^{-1} \vee \mathcal{L}(w^{(i)})$	Hessian

3.7 Optimality conditions

an PSD $\mathbf{H}(w^*) := \frac{\partial^2 \mathcal{L}(w^*)}{\partial w \partial w^T}$

4.1 Normal Equation

4 Least Squares

Necessary : $\nabla \mathcal{L}(w^*) = 0$ Sufficient : Hessi-

 $X^T(y - Xw) = 0 \Rightarrow w^*$

Cheat sheet

1 Regression

by Your Name, page 1 of 2

Simple $y_n \approx f(\mathbf{x_n}) := w_0 + w_1 \mathbf{x_{n1}}$

Multiple $y_n \approx f(\mathbf{x_n}) := w_0 + \sum_{j=1}^{D} w_j x_{nj} =$

1.1 Linear Regression

8.4 Nearest Neighbor Models Performs best in low dimensions. 8.4.1 k-NN $f_{S^{t,k}}(x) = \frac{1}{k} \sum_{n:x_n \in ngbh_{S^{t,k}(x)}} y_n$ Pick odd k so there is a clear winner. Large $k \rightarrow \text{large}$ bias small variance (inv.) 8.4.2 Error bound $\mathbb{E}[\mathcal{L}_{St}] \leq 2\mathcal{L}_{f^*} + 4c\sqrt{d}N^{-1/d+1}$ 8.5 Support Vector Machines (SVM) Logistic regression with hinge loss : $min_w \sum_{n=1}^{N} [1 - y_n x_n^T w]_+ + \frac{\lambda}{2} ||w||^2$ where $y \in [-1;1]$ is the label and hinge(x) = $max\{0,x\}$. Convex but not differentiable so need subgradient. duality : We can also use $\mathcal{L}(w) = max_{\alpha}G(w,\alpha)$. For SVM $min_w max_{\alpha \in [0,1]^N} \sum \alpha_n (1 - y_n x_n^T w) +$ $\frac{\lambda}{2}||w||^2$ differentiable and convex. Can switch max and min when convex in w and concave in α . This can make the formulation simpler: $w(\alpha) = \frac{1}{\lambda} \sum \alpha_n y_n x_n = \frac{1}{\lambda} X^T diag(y) \alpha$ $\max_{\alpha \in [0,1]^N} \alpha^T \mathbf{1} - \frac{1}{2\lambda} \alpha^T Y X X^T Y \alpha$ The lines that are lower bounds to the hingle $\partial f(g(w))\nabla g(w)$

 $H(w) = X^T S X$ with $S_{nn} = \sigma(x_n^T w)[1 -$

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

 $A(\eta) = ln[\int_{v} h(y)exp[\eta^{T}\psi(y)]dy]$

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

• $\eta_{gaussian} = (\mu/\sigma^2, -1/2\sigma^2)$

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

• $\eta_{bernoulli} = ln(\mu/1 - \mu)$

8.3 Exponential family

General form

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

Link function

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

• $\eta_{poisson} = ln(\mu)$

Cumulant

```
likelihood p(x_n|\theta) = \sum \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k).
                                                                              (number of parameters reduced from
                                                                              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)}}
                                                                                        \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)}}
                                                                                         \pi^{(t+1)} = \frac{1}{N} \sum_{n} q_{kn}^{(t)}
                                                                              9.3.2 General
which yields the optimisation problem: \theta^{(t+1)} := argmax_{\theta} \sum_{n=1}^{N} \mathbb{E}_{p(z_{n}|x_{n},\theta^{(t)})}[log \ p(x_{n},z_{n}|\theta)]^{\bullet} f(x) = -x^{3}, x \in [-1,0]
                                                                             10 Ouick maff
 solution is sparse (\alpha_n is the slope of the Chain rule h = f(g(w)) \rightarrow \partial h(w) =
                                                                             Gaussian \mathcal{N}(y|\mu,\sigma^2)\frac{1}{\sqrt{2\pi\sigma^2}}exp(-\frac{(y-\mu)^2}{\sigma^2})
                                                                             Multivariate Gaussian \mathcal{N}(y|\mu,\sigma^2)\frac{1}{\sqrt{(2\pi)^D det(\Sigma)}}exp(-\frac{1}{2}(y - 12.3 Non-convex functions • f(x)=x^3, x\in[-1,1]
```

9 Unsupervised Learning

 $min\mathcal{L}(z,\mu) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} ||x_n - \mu_k||_2^2$

with $z_{nk} \in \{0,1\}$ (unique assignments:

compute

 $\int 1 \text{ if } k = argmin_i ||x_n - \mu||^2$

Marginal likelihood: z_n are latent varia-

bles so they can be factored out from the

Algorithm (Coordinate Descent)

2. $\forall k$ compute $\mu_k = \frac{\sum_n z_{nk} x_n}{\sum_{k=1}^{n} z_{nk}}$

0 otherwise

re $pi_k = p(z_n = k)$

9.1 K-means clustering

 $\sum_{k} z_{nk} = 1).$

• $p(y) \rightarrow \text{prior}$ • $p(y|x) \rightarrow$ marginal likelihood • $p(x) \rightarrow \text{posterior}$ 10.1 Algebra $(PQ + I_N)^{-1}P = P(QP + I_M)^{-1}$ 11.2 $\stackrel{\hat{\mathsf{MAP}}}{\mathsf{solution}} = \sum_k \sum_n \frac{1}{2\sigma_k^2} (y_{nk} - x_n^T w_k)^2 +$ $\frac{1}{2}\sum_{k}||w_{k}||_{2}^{2}$ \rightarrow Likelihood p(y|X,w) = $\prod_{n}\prod_{k}\mathcal{N}(y_{nk}|w_{\iota}^{T}x_{n},\sigma_{\iota}^{2})$ and $p(w) = \prod_k \mathcal{N}(w_k|0, I_D)$ 11.3 Convexity $ln[\sum_{k=0}^{K} e^{t_k}]$ is convex. Linear sum of parameters is convex. 12 Multiple Choice Notes

Regularization term sometimes

renders the min. problem into a

strictly concave/convex problem.

data cannot be linearly separated.

• k-NN can be applied even if the

• $g(x) := min_v f(x, y) \Rightarrow g(x) \le$

• $f(x) = g(h(x)), x \in \mathbb{R}, g, h \text{ convex}$

• $max0, x = max_{\alpha \in [0,1]} \alpha x$

• $min0, x = min_{\alpha \in [0,1]} \alpha x$

• $max_{y}g(x) \leq max_{y}f(x,y)$

• $max_x min_v f(x, y)$

12.2 Convex functions

• $f(x) = x^2, x \in \mathbb{R}$

• $f(x) = e^{-x}, x \in \mathbb{R}$

• $f(x) = ln(1/x), x \in \mathbb{R}^+$

 $min_v max_x f(x, y)$

f(x,y)

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

p(x|y)p(y) = p(y|x)p(x) where

• $p(x|y) \rightarrow \text{likelihood}$

Naming Joint distribution p(x, y) =

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

Issues 11 Mock Exam Notes 1. Heavy computation 11.1 Normal equation 2. Spherical clusters Unique if convex. $\frac{1}{\sigma_{\nu}^{2}}X(X^{T}w_{k}-y_{k})+w_{k}=0 \Leftrightarrow$ 3. Hard clusters Probabilistic model $p(X|\mu,z) =$ $w_k^* = (\frac{1}{\sigma^2} X X^T + I_D)^{-1} \frac{1}{\sigma^2} X y_k$ $\textstyle\prod_{n}^{N}\mathcal{N}(x_{n}|\mu_{k},I)=\prod_{n}^{N}\prod_{k}^{K}\mathcal{N}(x_{n}|\mu_{k},I)^{z_{nk}}$ 9.2 Gaussian Mixture Models $p(X|\mu,z) = \prod_{n=1}^{N} p(x_n|z_n,\mu_k,\Sigma_k)p(z_n|\pi) =$ $\prod_{n=1}^{N}\prod_{k=1}^{K}[\mathcal{N}(x_{n}|\mu_{k},\Sigma_{k})]^{z_{nk}}\prod_{k=1}^{K}[\pi_{k}]^{z_{nk}}$ whe-

12.1 True statements

8.6 Kernel Ridge Regression From duality $w^* := X^T \alpha^*$ where $\alpha^* := (K +$

loss).

 $(\lambda I_N)^{-1}y$ and $K = XX^T = \phi^T(x)\phi(x) =$ $\kappa(x, x')$ (needs to be PSD and symmetric). μ)^T $\Sigma^{-1}(y - \mu)$)

• $f(x) = e^{-x^2}$, $x \in \mathbb{R}$

and increasing over R