1 Regression
1.1 Linear Regression
Simple $y_n \approx f(\mathbf{x_n}) := w_0 + w_1 \mathbf{x_{n1}}$
Multiple $y_n \approx f(\mathbf{x_n}) := w_0 + \sum_{j=1}^D w_j x_{nj} = \tilde{\mathbf{x}}_n^T \mathbf{w}$ If $D > N$ the task is under-determined (more dimensions than data) $\rightarrow$ regularization.

2 Cost functions
MSE $= \frac{1}{N}\sum_{n=1}^N [y_n - f(x_n)]^2$ Not good with outliers. MAE $= \frac{1}{N}\sum_{n=1}^N |y_n - f(x_n)|$

2.1 Convexity
A line joining two points never intersects with the function anywhere else. $f(\lambda u + (1-\lambda)v) \le \lambda f(u) + (1-\lambda) f(v)$ with $\lambda \in [0;1]$. A strictly convex function has a unique global minimum w^*. Sums of convex functions are convex.
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 the line segment between any two points of $\mathcal{C}$ lies in $\mathcal{C}$: $\theta u + (1-\theta)v \in \mathcal{C}$

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

3.1 Gradient descent
$w^{(t+1)} = w^{(t)} - \gamma\nabla\mathcal{L}(w^{(t)})$. Very sensitive to ill-conditioning.
GD - Linear Reg
$\mathcal{L}(w) = \frac{1}{2N}(y-Xw)^T(y-Xw) \rightarrow \nabla\mathcal{L}(w) = -\frac{1}{N}X^T(y-Xw)$.
Cost : $O_{error}(N*D) = 2N*D+N$ and $O_{weights} = 2N*D+D$.

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

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

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 : $h(e) = |e| \rightarrow g(e) = sgn(e)$ if $e \ne 0, \lambda$ otherwise . We get the gradient : $\nabla\mathcal{L}_{MAE} = -\frac{1}{N}\sum_n sgn(x_n)\nabla f(x_n)$.

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

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

4 Least Squares
4.1 Normal Equation
$X^T(y - Xw) = 0 \Rightarrow w^* = (XX^T)^{-1}X^Ty$ 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).

5 Likelihood
Probability of observing the data given a set of parameters and inputs : $p(y|X,w) = \prod p(y_n|x_n,w) = \prod\mathcal{N}(y_n|x_n^T w, \sigma^2)$
Best model maximises log-likelihood $\mathcal{L}_{LL} = -\frac{1}{2\sigma^2}\sum(y_n - x_n^T w)^2 + cst$.

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 : $w^*_{ridge} = argmin_w - log(p(w|X,y))$

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.

8 Classification
8.1 Optimal
$\hat{y}(x) = argmax_{y\in\mathcal{Y}} p(y|x)$

8.2 Logistic regression
$\sigma(z) = \frac{e^z}{1+e^z}$ to limit the predicted values $y \in [0;1]$ ($p(1|x) = \sigma(x^Tw)$ and $p(0|x) = 1-\sigma(x^Tw)$).
Likelihood
$p(y|X,w) = \prod p(y_n|x_n) = \prod_{n:y_n=0} p(y_n = 0|x_n)...\prod_{n:y_n=K} p(y_n = K|x_n) = \prod_k^K\prod_n^N [p(y_n = k|x_n,w)]^{\tilde{y}_{nk}}$ where $tildey_{nk} = 1$ if $y_n = k$.
For binary classification
$p(y|X,w) = \prod p(y_n|x_n) = \prod_{n:y_n=0} p(y_n = 0|x_n)\prod_{n:y_n=1} p(y_n = 1|x_n) = \prod_n^N \sigma(x_n^Tw)^{y_n}[1-\sigma(x_n^Tw)]^{1-y_n}$
Loss
$\mathcal{L}(w) = \sum_{n=1}^N ln(1+exp(x_n^Tw)) - y_n x_n^T w$ 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
$p(y|\eta) = h(y)exp[\eta^T\psi(y) - A(\eta)]$ where
Cumulant
$A(\eta) = ln[\int_y h(y)exp[\eta^T\psi(y)]dy]$
$\nabla A(\eta) = \mathbb{E}[\psi(y)]$
$\nabla^2 A(\eta) = \mathbb{E}[\psi\psi^T] - \mathbb{E}[\psi]\mathbb{E}[\psi^T]$
Link function
$\eta = g^{-1}(\mu) \Leftrightarrow \mu = g(\eta)$

- $\eta_{gaussian} = (\mu/\sigma^2, -1/2\sigma^2)$
- $\eta_{poisson} = ln(\mu)$
- $\eta_{bernoulli} = ln(\mu/1-\mu)$
- $\eta_{general} = g^{-1}(\frac{1}{N}\sum_{n=1}^N\psi(y_n))$

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 $k \rightarrow$ 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 : $min_w\sum_{n=1}^N[1-y_nx_n^Tw]_+ + \frac{\lambda}{2}\|w\|^2$ where $y\in[-1;1]$ is the label and $hinge(x) = max\{0,x\}$. Convex but not differentiable so need subgradient.
We can also use duality : $\mathcal{L}(w) = max_\alpha G(w,\alpha)$. For SVM $min_w max_{\alpha\in[0,1]^N}\sum\alpha_n(1-y_nx_n^Tw) + \frac{\lambda}{2}\|w\|^2$ differentiable and convex.
Can switch max and min when convex in w and concave in α. This can make the formulation simpler:
$w(\alpha) = \frac{1}{\lambda}\sum\alpha_ny_nx_n = \frac{1}{\lambda}X^Tdiag(y)\alpha$ 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 hingle 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. $\forall n$ compute $z_n = \begin{cases}1 \text{ if } k = argmin_j\|x_n-\mu\|^2 \\ 0 \text{ otherwise}\end{cases}$
2. $\forall k$ compute $\mu_k = \frac{\sum_n z_{nk}x_n}{\sum_n z_{nk}}$

Issues

1. Heavy computation
2. Spherical clusters
3. Hard clusters

Probabilistic model $p(X|\mu,z) = \prod_n^N\mathcal{N}(x_n|\mu_k,I) = \prod_n^N\prod_k^K\mathcal{N}(x_n|\mu_k,I)^{z_{nk}}$

9.2 Gaussian Mixture Models
$p(X|\mu,z) = \prod_n^N p(x_n|z_n,\mu_k,\Sigma_k)p(z_n|\pi) = \prod_n^N\prod_k^K[\mathcal{N}(x_n|\mu_k,\Sigma_k)]^{z_{nk}}\prod_k^K[\pi_k]^{z_{nk}}$ where $pi_k = p(z_n = k)$
Marginal likelihood: z_n are latent variables so they can be factored out from the 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
Initialize $\mu^{(1)},\Sigma^{(1)},\pi^{(1)}$.

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

10 Quick maff
Chain rule $h = f(g(w)) \rightarrow \partial h(w) = \partial f(g(w))\nabla g(w)$
Gaussian $\mathcal{N}(y|\mu,\sigma^2)\frac{1}{\sqrt{2\pi\sigma^2}}exp(-\frac{(y-\mu)^2}{\sigma^2})$
Multivariate Gaussian $\mathcal{N}(y|\mu,\sigma^2)\frac{1}{\sqrt{(2\pi)^Ddet(\Sigma)}}exp(-\frac{1}{2}(y-\mu)^T\Sigma^{-1}(y-\mu))$
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

- $p(x|y) \rightarrow$ likelihood
- $p(y) \rightarrow$ prior
- $p(y|x) \rightarrow$ marginal likelihood
- $p(x) \rightarrow$ posterior

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

11 Mock Exam Notes
11.1 Normal equation
Unique if convex.
$\frac{1}{\sigma_k^2}X(X^Tw_k - y_k) + w_k = 0 \Leftrightarrow w_k^* = (\frac{1}{\sigma_k^2}XX^T + I_D)^{-1}\frac{1}{\sigma_k^2}Xy_k$

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

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

12 Multiple Choice Notes
12.1 True statements
- Regularization 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.
- $max0,x = max_{\alpha\in[0,1]}\alpha x$
- $min0,x = min_{\alpha\in[0,1]}\alpha x$
- $g(x) := min_y f(x,y) \Rightarrow g(x) \le f(x,y)$
- $max_x g(x) \le max_x f(x,y)$
- $max_x min_y f(x,y) \le min_y max_x f(x,y)$

12.2 Convex functions
- $f(x) = x^2, x\in\mathbb{R}$
- $f(x) = -x^3, x\in[-1,0]$
- $f(x) = e^{-x}, x\in\mathbb{R}$
- $f(x) = ln(1/x), x\in\mathbb{R}^+$
- $f(x) = g(h(x)), x\in\mathbb{R}, g,h$ convex and increasing over $\mathbb{R}$

12.3 Non-convex functions
- $f(x) = x^3, x\in[-1,1]$
- $f(x) = e^{-x^2/2}, x\in\mathbb{R}$