# 10-715 Advanced Introduction to Machine Learning

**Homework 1 Solutions** *October 6, 2014*

## 1 Regression (Samy)

### 1.1 Multi-Task Regresion

1. The Cost function can be decomposed as,

$$J_0(\Theta) = \sum_{j=1}^{k} \|\mathbf{Y}_{j,:} - \Theta_{j,:}X\|^2$$

where $\mathbf{Y}_{j,:}, \Theta_{j,:}$ refer to the $j^{th}$ rows of $\mathbf{Y}, \Theta$ respectively. Since this essentially decouples the parameters involved with each task, we can solve them separately.

2. (a) Independent: Yes. Convex: Yes.
This is the usual $L_2$ regularization to control the variance.

   (b) Independent: No. Convex: Yes.
Here we are trying to model all our outputs as a function of a sparse subset of the covariates.

   (c) Independent: No. Convex: No.
Here, by encouraging $\Theta$ to be low rank we are trying to create (linear) dependence across multiple tasks.
e.g. Say we are trying to predict precipitation in different regions based on different weather features. We want different models for each region since a universal model may not be suitable. However, all these tasks are likely to be related and so we want to encourage dependence. In doing so, we reduce the sample complexity of learning all tasks since data from one region will be useful in estimating the parameters of another region.
Some of you also pointed out that a rank penalty is intractable. This is true. A commonly used convex relaxation is to use a nuclear norm penalty.

### 1.2 Shrinkage in Ridge Regression

1. The solution to the Ridge Regression problem is $\widehat{\beta} = (X^\top X + \lambda I)^{-1} X^\top y$. Using the SVD $X = U\Sigma V^\top$,

$$\begin{aligned}
\widehat{\beta} &= V(\Sigma^2 + \lambda I)^{-1}\Sigma U^\top y \\
x_*^\top \widehat{\beta} &= z_*^\top V^\top \widehat{\beta} \\
&= z_*^\top (\Sigma^2 + \lambda I)^{-1}\Sigma U^\top y \\
&= \sum_{i=1}^{d} \frac{z_{*i}\sigma_i}{\sigma_i^2 + \lambda} u_i^\top y
\end{aligned}$$

2. Since $X$ has zero mean, the directions $v_1, \ldots, v_n$ are the eigenvectors of the empirical covariance matrix. The expression $\frac{z_{*i}\sigma_i}{\sigma_i^2+\lambda}$ indicates that the directions along which the empirical covariance is lowest are shrinked the most. In the directions where data is more spread out (empirical covariance high) we can estimate the gradients of our linear function well since it would be less susceptible to noise. In the directions where there is less spread , there is high variance in the estimate of the gradient. Ridge

regression helps us control the variance by imposing different penalties along different principal axes. Some of you also made the equivalent argument that if $X$ was poorly conditioned then it would blow up the variance in the directions in which $\sigma_i$ was small. The penalty prevents this from happening.

## 1.3 Local/Weighted Linear Regression

1. Using the given notation, we can express $\widehat{\beta}$ as follows,

$$\widehat{\beta} = \underset{\beta}{\operatorname{argmin}} \|W^{1/2}(\mathbf{y} - X\beta)\|^2$$

By settings its gradient to zero we get $\widehat{\beta} = (X^\top W X)^{-1} X^\top W \mathbf{y}$. Substituting $\hat{f}(x) = x^\top \widehat{\beta}$ yields the required answer.

2. By setting $\beta = \theta$, $x = 1$, $\implies X = \mathbf{1}^\top \in \mathbb{R}^n$ and $\hat{f}(x) = \beta^\top x = \theta$ we get the same problem as above. Then $(X^\top W X) = \sum_i w_i$, $X^\top W \mathbf{y} = \sum_i w_i y_i$ which yields

$$\hat{f}(x) = \frac{\sum_i w_i(x) y_i}{\sum_i w_i} = \sum_i \alpha_i y_i$$

where $\alpha_i = w_i / \sum_j w_j$. For $w_i(x) = k((x - x_i)/h)$ we get precisely the Nadaraya-Watson Estimator. Since the prediction at any point is a convex combination of the observed labels it always lies in between the maximum and the minimum.

## 1.4 Least Norm Solution

1. The solution may be obtained by solving the problem,

$$\text{minimize } \|\beta\|^2, \qquad \text{subject to } X\beta = \mathbf{y}$$

The Lagrangian for the problem is $\mathcal{L}(\nu, \beta) = \beta^\top \beta + \nu^\top (X\beta - yb)$. By setting $\nabla_\beta \mathcal{L} \triangleq 0$ and then substituting back we get, $\widehat{\beta}_{ln} = X^\top (X X^\top)^{-1} \mathbf{y}$.

2. Let $J = \|y - X\beta\|^2$ and $\beta_* = X^\dagger y$. $\nabla_\beta J = 2X^\top X\beta - 2X^\top y$. When $\beta = \beta_*$, $\nabla_\beta J = 2X^\top X X^\dagger y - 2X^\top y = 2X^\top y - 2X^\top y = 0$. Since $J$ is convex in $\beta$ and $\beta_*$ satisfies the stationarity condition we have that $\beta_*$ is a least squares solution.

Let $\beta$ be any other least squares solution: $\|y - X\beta\| = \|y - X\beta_*\|$. Then,

$$\|y - X\beta\|^2 = \|y - X\beta_* - X\delta\|^2 = \|y - X\beta_*\|^2 + \|X\delta\|^2$$

The last step follows by observing that $(y - X\beta_*)^\top X\delta = y^\top X\delta - y^\top (X^\dagger)^\top X^\top X\delta = 0$. Hence $\delta \in \mathcal{N}(X)$ and $\beta_* \perp \delta$. Therefore $\|\beta\|^2 = \|\beta_*\|^2 + \|\delta\|^2 \leq \|\beta_*\|^2$.

Some of you presented alternative arguments, mostly based on the SVD characterization of the MP inverse.

# 2 Pólya Discriminant Analysis (Samy)

## 2.1 Model

1. Conditioned on $y_i = k, m_i$ the distribuiton of $x$ corresponds to a Dirichlet Multinomial with parameters $m_i, \alpha_k$. Its mass function and the logarithm is

$$p_{dm}(x_i; \alpha_k) = \frac{\Gamma(A_k)}{\Gamma(m_i + A_k)} \prod_{s=1}^{V} \frac{\Gamma(x_i^{(s)} + \alpha_k^{(s)})}{\Gamma(\alpha_k^{(s)})}$$

$$\log p_{dm}(x_i; \alpha_k) = \log \Gamma(A_k) - \log \Gamma(m_i + A_k) + \sum_{s=1}^{V} \Big( \log \Gamma(x_i^{(s)} + \alpha_k^{(s)}) - \log \Gamma(\alpha_k^{(s)}) \Big)$$

where $A_k = \sum_{s=1}^{V} \alpha_k^{(s)}$ and $m_i = \sum_{s=1}^{V} x_i^{(s)}$.

The likelihood and log likelihood of the data $\mathcal{D} = (x_i, y_i)_{i=1}^n$ is then given by,

$$p(\mathcal{D}; \theta, \alpha_1, \ldots, \alpha_K) = \prod_{i=1}^{n} \prod_{k=1}^{K} \Big( \theta^{(k)} p_{dm}(x_i; \alpha_k) \Big)^{\mathbb{1}(y_i = k)}$$

$$\ell(\theta, \alpha_1, \ldots, \alpha_K; \mathcal{D}) = \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{1}(y_i = k) \Big( \log \theta^{(k)} + \log p_{dm}(x_i; \alpha_k) \Big)$$

2. We need to maximize the above log likelihood w.r.t $\theta$ subject to the constraint $\sum_k \theta^{(k)} = 1$. The corresponding Lagrangian is,

$$\mathcal{L} = \sum_{k=1}^{K} \log \theta^{(k)} \sum_{i=1}^{n} \mathbb{1}(y_i = k) + \nu \left( \sum_{k=1}^{K} \theta^{(k)} - 1 \right)$$

Solving this for $\theta^{(k)}$ yields the MLE estimate

$$\hat{\theta}^{(k)} := \frac{\sum_{i=1}^{n} \mathbb{1}(y_i = k)}{n} = \frac{\#\text{ training instances in class } k}{n}$$

3. The solution to this part are based on ideas from [Min00]. The first and second partial derivatives of the log likelihood are,

$$\frac{\partial \ell}{\partial \alpha_k^{(s)}} = \sum_{i=1}^{n} \mathbb{1}(y_i = k) \Big( \Psi(A_k) - \Psi(m_i + A_k) + \Psi(x_i^{(s)} + \alpha_k^{(s)}) - \Psi(\alpha_k^{(s)}) \Big)$$

$$\frac{\partial^2 \ell}{\partial {\alpha_k^{(s)}}^2} = \sum_{i=1}^{n} \mathbb{1}(y_i = k) \Big( \Psi'(A_k) - \Psi'(m_i + A_k) + \Psi'(x_i^{(s)} + \alpha_k^{(s)}) - \Psi'(\alpha_k^{(s)}) \Big)$$

$$\frac{\partial^2 \ell}{\partial \alpha_k^{(s)} \partial \alpha_k^{(t)}} = \sum_{i=1}^{n} \mathbb{1}(y_i = k) \Big( \Psi'(A_k) - \Psi'(m_i + A_k) \Big)$$

where $\Psi, \Psi'$ are the di-gamma and tri-gamma functions respectively. The gradient $\mathbf{g}_k \in \mathbb{R}^V$ for optimizing $\alpha_k$ is given by $\mathbf{g}_k^{(s)} = \frac{\partial \ell}{\partial \alpha_k^{(s)}}$. The Hessian $\mathbf{H}_k \in \mathbb{R}^{V \times V}$ can be written as, $\mathbf{H}_k = \mathbf{D} + z\mathbf{1}\mathbf{1}^\top$ where $\mathbf{D}$ a diagonal matrix and $z \in \mathbb{R}$ are given by,

$$\mathbf{D}_{ss} = \sum_{i \in [k]} \Psi'(x_i^{(s)} + \alpha_k^{(s)}) - n_k \Psi'(\alpha_k^{(s)})$$

$$z = n_k \Psi'(A_k) - \sum_{i \in [k]} \Psi'(m_i + A_k)$$

Here $[k]$ refers to the set of training instances in class $k$ and $n_k = |[k]|$. By the Sherman Morrison formula, $\mathbf{H}_k^{-1}$ can be computed as

$$\mathbf{H}_k^{-1} = \mathbf{D}^{-1} - \frac{\mathbf{D}^{-1}\mathbf{1}\mathbf{1}^\top\mathbf{D}^{-1}}{1/z + \mathbf{1}^\top\mathbf{D}^{-1}\mathbf{1}}$$

The Newton's method update is then given by $\alpha_k^{new} \leftarrow \alpha_k^{old} - \mathbf{H}_k^{-1}\mathbf{g}_k$.

To analyse the complexity, note that we first need to compute and store $\mathbf{g}_k$, $D$ and $z$. This requires only $\tilde{O}(V)$ time and space complexity. Since we can write,

$$[\mathbf{H}_k^{-1}\mathbf{g}_k]^{(i)} = \mathbf{g}_k^{(i)}/\mathbf{D}^{(i,i)} - \frac{\sum_j \mathbf{g}_k^{(j)}/\mathbf{D}^{(j,j)}}{1/z + \sum_j 1/\mathbf{D}^{(j,j)}} \left(1/\mathbf{D}^{(i,i)}\right)$$

the inversion can be done in $\tilde{O}(V)$ time.

4. We choose the class that maximizes the posterior $p(y|x) = \frac{p(x|y)p(y)}{p(x)}$,

$$y_* = \underset{k\in\{1,\ldots,K\}}{\operatorname{argmax}}\ p(y=k|x_*) = \underset{k\in\{1,\ldots,K\}}{\operatorname{argmax}}\ p(x_*|y=k)p(y=k) = \underset{k\in\{1,\ldots,K\}}{\operatorname{argmax}}\ p_{dm}(x_*;\alpha_k)\theta^{(k)}$$

5. In the given Bayesian formulation, we can write the joint and log-joint probability as,

$$\begin{aligned}
p(\mathcal{D},\theta,\alpha_1,\ldots,\alpha_K) &= p(\theta)\left(\prod_{k=1}^K p(\alpha_k)\right) p(\mathcal{D}|\theta,\alpha_1,\ldots,\alpha_K) \\
&= \left(\frac{\Gamma(\sum_k \theta_0^{(k)})}{\prod_k \Gamma(\theta_0^{(k)})}\prod_{k=1}^K \theta^{(k)^{\theta_0^{(k)}-1}}\right)\left(\prod_{k=1}^K \frac{1}{(2\pi)^{K/2}(1/2\lambda)^{K/2}}\exp\left(-\lambda\|\alpha_k\|^2\right)\right) p(\mathcal{D}|\theta,\alpha_1,\ldots,\alpha_K) \\
\tilde{\ell}(\mathcal{D},\theta,\alpha_1,\ldots,\alpha_K) &= \sum_{k=1}^K(\theta_0^{(k)}-1)\log\theta^{(k)} + \sum_{k=1}^K -\lambda\|\alpha_k\|^2 + \ell(\theta,\alpha_1,\ldots,\alpha_K;\mathcal{D}) + C(\theta_0,\lambda)
\end{aligned}$$

where $C(\theta_0,\lambda)$ is a constant term.
As before, by writing out the Lagrangian and optimizing for $\theta$ we get,

$$\hat{\theta}^{(k)} := \frac{n_k + \theta_0^{(k)} - 1}{n + \sum_j \theta_0^{(j)} - K}$$

As for $\alpha_k$, the partial derivatives are ,

$$\frac{\partial\tilde{\ell}}{\partial\alpha_k^{(s)}} = \frac{\partial\ell}{\partial\alpha_k^{(s)}} - 2\lambda\alpha_k, \qquad \frac{\partial^2\tilde{\ell}}{\partial{\alpha_k^{(s)}}^2} = \frac{\partial^2\ell}{\partial{\alpha_k^{(s)}}^2} - 2\lambda \qquad \frac{\partial^2\tilde{\ell}}{\partial\alpha_k^{(s)}\partial\alpha_k^{(t)}} = \frac{\partial^2\ell}{\partial\alpha_k^{(s)}\partial\alpha_k^{(t)}}$$

We can perform the Newton's step efficiently using the same trick by setting $z$ to be the same as before and

$$\begin{aligned}
\mathbf{g}_k^{(s)} &= n_k\Psi(A_k) - \sum_{i=1}^n \Psi(m_i + A_k) + \sum_{i=1}^n \Psi(x_i^{(s)} + \alpha_k^{(s)}) - n_k\Psi(\alpha_k^{(s)}) - 2\lambda\alpha_k \\
\mathbf{D}_{ss} &= \sum_{i\in[k]} \Psi'(x_i^{(s)} + \alpha_k^{(s)}) - n_k\Psi'(\alpha_k^{(s)}) - 2\lambda
\end{aligned}$$

## 2.2 Experiment

This is our Matlab implementation.

```
function [theta, alpha] = trainPDA(X, y, theta0, lambda)

  % Prelims
  K = numel(unique(y));
  V = size(X, 2);
  numData = size(X, 1);

  % MAP for theta
  table = tabulate(y);
  adjustedFreqs = table(:,2) + theta0 - 1;
  theta = adjustedFreqs/sum(adjustedFreqs);

  % MAP for alpha
  alpha = zeros(K, V);
  for k = 1:K
    Xk = X( y==k, :);
    alpha(k, :) = newtonRaphsonPDA(Xk, lambda);
  end
end


function [alpha_k] = newtonRaphsonPDA(Xk, lambda)

  % Prelims
  numNRIters = 10; % Just use 5 iterations of NR
  nk = size(Xk, 1); % number of training data in this class
  m = sum(Xk, 2); % number of words in each documents
  initPt = sum(Xk); initPt = initPt/sum(initPt); % Initialization

  % Now perform Newton's
  alpha_k = initPt; % alphak in the current iteration
  for nrIter = 1:numNRIters
    % Compute the following
    Ak = sum(alpha_k);
    XplusAlpha = bsxfun(@plus, Xk, alpha_k);
    % The gradient
    g = nk * psi(Ak) - sum(psi(m + Ak)) + sum( psi(XplusAlpha) ) ...
        - nk * psi(alpha_k) - 2 * lambda * alpha_k;
    % The value z ( see solutions)
    z = nk * psi(1, Ak) - sum(psi(1, m + Ak));
    % The diagonal of the Hessian
    D = sum(psi(1, XplusAlpha)) - nk * psi(1, alpha_k) - 2*lambda;
    % Newton's step update
    Hinvg = g./D - (1./D) * sum(g./D) / (1/z + sum(1./D));
    alpha_k = alpha_k - 1*Hinvg;
  end
end


function logL = classLogLikelihoods(X, alphak)
```

```
  % Prelims
  Ak = sum(alphak);
  m = sum(X, 2); % number of words in each documents
  XplusAlpha = bsxfun(@plus, X, alphak);

  % Compute the log likelihood
  logL = gammaln(Ak) - gammaln(m + Ak) + ...
    sum(gammaln(XplusAlpha), 2) - sum( gammaln(alphak) );

end


function [preds, classLogJoints] = predictPDA(X, theta, alpha)

  % prelims
  n = size(X, 1);
  K = numel(theta);

  % First obtain the class log joint probabilities
  classLogLs = zeros(n, K);
  for k = 1:K
    classLogLs(:, k) = classLogLikelihoods(X, alpha(k, :));
  end
  classLogJoints = bsxfun(@plus, classLogLs, log(theta'));

  % Finally obtain the predictions
  [~, preds] = max(classLogJoints, [], 2);

end
```

# 3 Duality

## 3.1 Weak Duality

1. $L(x, \lambda, u) = f(x) + \lambda h_1(x) + u h_2(x)$
2. $g(\lambda, u) = \inf_{x \in \mathbb{R}^d} L(x, \lambda, u)$
3. Let $P$ denote the feasible region of the primal. If $x \in P$, that is, $h_1(x) \leq 0$, $h_2(x) = 0$, then for any $\lambda \geq 0, u \in R$, we have $f(x) \geq f(x) + \lambda h_1(x) + u h_2(x) = L(x, \lambda, u)$. Taking infimums over $x \in P$,

$$\inf_{x \in P} f(x) \geq \inf_{x \in P} L(x, \lambda, u) \geq \inf_{x \in \mathbb{R}^d} L(x, \lambda, u) = g(\lambda, u)$$

   The last inequality holds because $P \subseteq \mathbb{R}^d$. The required result follows from the observation that the above inequality holds for $\forall \lambda \geq 0, u$.

### 3.2 Optimal Coding

1. Let $P$ denote the feasible region. It is sufficient to show that $x^\alpha = \alpha x + (1-\alpha)y \in P$ given $x, y \in P$, for $\alpha \in (0,1)$. Using weighted AM-GM inequality and the feasibility of $x, y$, we can write

$$\sum_{i=1}^{n} 2^{-(\alpha x_i + (1-\alpha)y_i)} \leq \sum_{i=1}^{n} \alpha 2^{-x_i} + (1-\alpha)2^{-y_i} = \alpha \sum_{i=1}^{n} 2^{-x_i} + (1-\alpha)\sum_{i=1}^{n} 2^{-y_i} \leq \alpha + (1-\alpha) = 1$$

So $x^\alpha$ satisfies the first inequality constraint. Further it is clear that $x_i, y_i \geq 0 \Rightarrow x_i^\alpha \geq 0$. So $x^\alpha \in P$, which proves the convexity of P.

2. Suppose for the purpose of contradiction that an optimal solution $x^*$ satisfies the strict inequality, that is, $\sum_{i=1}^{n} 2^{-x_i^*} < 1$. $\forall j, x_j^* > 0$, because otherwise, $\sum_{i=1}^{n} 2^{-x_i^*} > 1$. So, one of the $x_j$'s can be reduced so that the objective is reduced while still maintaining feasibility. This means $x^*$ is not an optimal solution, which is a contradiction.

3. For $\lambda \in \mathbb{R}, u_i \geq 0$, the Lagrange function is

$$L(x, \lambda, u) = \sum_{i=1}^{n} p_i x_i + \lambda(\sum_{i=1}^{n} 2^{-x_i} - 1) - \sum_{i=1}^{n} u_i x_i \tag{1}$$

Let $(x, \lambda, u)$ satisfy KKT conditions. From complementary slackness, $\forall i \in [n]$, we have $u_i x_i = 0$. As shown in the previous part, $x_i > 0$ for any feasible point, which means $u_i = 0$. $L(x, \lambda, u)$ is convex in $x$ as the first and third terms in (1) are linear and the Hessian of the second term is positive definite. So the stationarity condition $0 \in \partial L(x)$ becomes $0 = \nabla L(x)$ when $L$ is treated as a function of $x$ alone. $\forall i \in [n]$, as $u_i = 0$

$$\frac{\partial L}{\partial x_i} = p_i - \lambda 2^{-x_i} \log 2 - u_i = p_i - \lambda 2^{-x_i} \log 2.$$

$\frac{\partial L}{\partial x_i} = 0 \Leftrightarrow$

$$p_i = (\lambda \log 2) 2^{-x_i}. \tag{2}$$

Summing over $i$, and noting that $\sum_{i=1}^{n} 2^{-x_i} = 1$, we get

$$\sum_{i=1}^{n} p_i = \lambda \log 2 \sum_{i=1}^{n} 2^{-x_i}$$
$$1 = \lambda \log 2$$

So $\lambda = 1/\log 2$ and from (2), we have $p_i = 2^{-x_i}$ and hence $x_i = -\log_2 p_i$. It is easy to verify that $x_i = -\log_2 p_i, \lambda = 1/\log 2, u_i = 0$ satisfy the KKT conditions and hence it is the optima.

# 4 SVM and Perceptron (Veeru)

### 4.1

Start with the primal and write the KKT conditions. For notation, I will use equations (48)-(56) from Chris Burges tutorial on SVM(http://www.umiacs.umd.edu/ ramanath/svm.pdf).

$\alpha_i = 0 \overset{(50)}{\Rightarrow} \mu_i = C \overset{(56)}{\Rightarrow} \xi_i = 0 \overset{(51)}{\Rightarrow} y_i f(x_i) \geq 1$

$0 < \alpha_i < C \overset{(50),(55)}{\Rightarrow} \mu_i > 0, y_i f(x_i) - 1 + \xi_i = 0 \overset{(56)}{\Rightarrow} \xi_i = 0, y_i f(x_i) - 1 + \xi_i = 0 \Rightarrow y_i f(x_i) = 1.$

$\alpha_i = C \overset{(55)}{\Rightarrow} y_i f(x_i) - 1 + \xi_i = 0 \overset{(52)}{\Rightarrow} y_i f(x_i) \leq 1$

### 4.2

Let me know if you have any difficulty with this.

### 4.3 Mistake bound for Perceptron

Let $(x^k, y^k)$ be the datapoint for which the perceptron fails in the $k$th step, $k \in \mathbb{N}$. That is, $\langle w_{k-1}, y^k x^k \rangle < 0$. We have $w_k = w_{k-1} + y^k x^k$ from the algorithm.

1. Using this, and the fact that $\forall i \in [n], \langle y_i x_i, w_* \rangle \geq \delta$, we can write

$$\langle w_k, w_* \rangle = \langle w_{k-1}, w_* \rangle + \langle y^k x^k, w_* \rangle \geq \langle w_{k-1}, w_* \rangle + \delta$$

   Telescoping and using $w_0 = 0$, we get $\langle w_k, w_* \rangle \geq k\delta$.

2.

$$\begin{aligned} \|w_k\|^2 &= \|w_{k-1} + y^k x^k\|^2 \\ &= \|w_{k-1}\|^2 + 2\langle w_{k-1}, y^k x^k \rangle + \|y^k x^k\|^2 \\ &\leq \|w_{k-1}\|^2 + \|x^k\|^2 \\ &\leq \|w_{k-1}\|^2 + M^2 \end{aligned}$$

   We used $\langle w_{k-1}, y^k x^k \rangle < 0$ and $y^k = \pm 1$ to get the first inequality. Again telescoping and using $w_0 = 0$, we arrive at $\|w_k\|^2 \leq kM^2$.

3.

$$kM^2 \geq \|w_k\|^2 \geq \langle w_k, w_* \rangle^2 \geq k^2\delta^2.$$

   We used the second part in the first inequality and the first part in the third inequality. The second inequality is obtained by noting that $\|w_*\| = 1$ and using Cauchy-Schwartz inequality. From $kM^2 \geq k^2\delta^2$, it easily follows that $k \leq M^2/\delta^2$.

## References

[Min00] Thomas P. Minka. Estimating a Dirichlet Distribution. Technical report, 2000.