CS 189 Introduction to Machine Learning
Summer 2018 HW7

Your self-grade URL is http://eecs189.org/self_grade?question_ids=1_1,1_2,2_1,2_2,2_3,3_1,3_2,3_3,3_4,3_5,3_6,4_1,4_2,4_3.

This homework is due **Monday, August 6 at 10pm.**

2 SVM with custom margins

In the lecture, we covered the soft margin SVM. The objective to be optimized over the training set $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$ is

$$\min_{\mathbf{w},b,\xi_i} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n} \xi_i \tag{1}$$

$$s.t. \quad y_i(\mathbf{w}^\top \mathbf{x}_i - b) \geq 1 - \xi_i \quad \forall i \tag{2}$$

$$\xi_i \geq 0 \quad \forall i \tag{3}$$

In this problem, we are interested in a modified version of the soft margin SVM where we have a custom margin for each of the n data points. In the standard soft margin SVM, we pay a penalty of ξ_i for each of the data point. In practice, we might not want to treat each training point equally, since with prior knowledge, we might know that some data points are more important than the others. There is some connection to weighted least squares. We formally define the following optimization problem:

$$\min_{\mathbf{w},b,\xi_i} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n} \phi_i\xi_i \tag{4}$$

$$s.t. \quad y_i(\mathbf{w}^\top \mathbf{x}_i - b) \geq 1 - \xi_i \quad \forall i \tag{5}$$

$$\xi_i \geq 0 \quad \forall i \tag{6}$$

Note that the only difference is that we have a weighting factor $\phi_i > 0$ for each of the slack variables ξ_i in the objective function. ϕ_i are some constants given by the prior knowledge, thus they can be treated as known constants in the optimization problem. Intuitively, this formulation weights each of the violations (ξ_i) differently according to the prior knowledge (ϕ_i).

(a) For the standard soft margin SVM, we have shown that the constrained optimization problem is equal to the following unconstrained optimization problem, i.e. regularized empirical risk minimization problem with hinge loss:

$$\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n} \max(1 - y_i(\mathbf{w}^\top \mathbf{x}_i - b), 0) \tag{7}$$

What's the corresponding unconstrained optimization problem for the SVM with custom margins?

Solution: The corresponding unconstrained optimization problem is

$$\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n} \phi_i \max(1 - y_i(\mathbf{w}^\top \mathbf{x}_i - b), 0). \tag{8}$$

We can see this as follows. Manipulating the first inequality, we have that

$$\xi_i \geq 1 - y_i(\mathbf{w}^\top \mathbf{x}_i - b) \quad \forall i. \tag{9}$$

Combining this with the second inequality, we have that

$$\xi_i \geq \max(1 - y_i(\mathbf{w}^\top \mathbf{x}_i - b), 0). \tag{10}$$

Since we are minimizing and since we know that $\phi_i > 0$ for all i, we conclude that the constraint must be tight:

$$\xi_i = \max(1 - y_i(\mathbf{w}^\top \mathbf{x}_i - b), 0). \tag{11}$$

The above unconstrained problem then follows when we substitute for ξ_i.

(b) The dual of the standard soft margin SVM is:

$$\max_{\alpha} \alpha^\top \mathbf{1} - \frac{1}{2}\alpha^\top \mathbf{Q}\alpha \tag{12}$$

$$s.t. \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{13}$$

$$0 \leq \alpha_i \leq C \quad i = 1, \cdots, n \tag{14}$$

where $\mathbf{Q} = (\operatorname{diag} \mathbf{y})\mathbf{X}\mathbf{X}^T(\operatorname{diag} \mathbf{y})$

What's the dual form of the SVM with custom margin? Show the derivation steps in detail.

Solution: We start from the constrained primal problem.

$$\min_{\mathbf{w},b,\xi_i} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n} \phi_i \xi_i \tag{15}$$

$$s.t. \quad y_i(\mathbf{w}^\top \mathbf{x}_i - b) \geq 1 - \xi_i \quad \forall i \tag{16}$$

$$\xi_i \geq 0 \quad \forall i \tag{17}$$

Using α and β for our dual variables, the Lagrangian is then

$$\mathcal{L}(\mathbf{w}, b, \xi, \alpha, \beta) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\phi_i\xi_i + \sum_{i=1}^{n}\alpha_i(1 - \xi_i - y_i(\mathbf{w}^\top\mathbf{x}_i - b)) + \sum_{i=1}^{n}\beta_i(-\xi_i) \tag{18}$$

$$= \frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\alpha_i y_i(\mathbf{w}^\top\mathbf{x}_i - b) + \sum_{i=1}^{n}(C\phi_i - \alpha_i - \beta_i)\xi_i \tag{19}$$

The optimization we want to solve then is

$$\max_{\alpha\geq 0, \beta\geq 0}\min_{\mathbf{w},b,\xi_i}\mathcal{L}(\mathbf{w}, \mathbf{b}, \xi, \alpha, \beta).$$

Since the problem is convex and strictly feasible, we know that the KKT conditions must hold for the dual optimal solutions. We will now use the KKT conditions to simplify our problem. First, we know that the gradients with respect to all primal variables must be 0 by the stationarity condition. From this, we have that

$$\nabla_{\mathbf{w}}\mathcal{L} = \mathbf{w}^* - \sum_{i=1}^{n}\alpha_i^* y_i x_i = 0 \implies \mathbf{w}^* = \sum_{i=1}^{n}\alpha_i^* y_i x_i.$$

$$\nabla_b\mathcal{L} = \sum_{i=1}^{n}\alpha_i^* y_i = 0.$$

$$\nabla_{\xi_i}\mathcal{L} = C\phi_i - \alpha_i^* - \beta_i^* = 0 \quad i = 1, \ldots, n.$$

Since α, β are restricted to being greater than or equal to 0, the last equality implies that $\alpha_i^* \leq C\phi_i$. Now using the equations above, we can simplify the Lagrangian to

$$\mathcal{L}(\mathbf{w}, b, \xi, \alpha^*, \beta^*) = \frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^{n}\alpha_i^* - \sum_{i=1}^{n}\alpha_i^* y_i \mathbf{w}^\top\mathbf{x}_i.$$

We need to minimize the above function then to find the optimal $\mathbf{w}^*, b^*, \xi^*$.

$$\min_{\mathbf{w},b,\xi}\mathcal{L}(\mathbf{w}, b, \xi, \alpha^*, \beta^*) = \mathcal{L}(\mathbf{w}^*, b^*, \xi^*, \alpha^*, \beta^*) \tag{20}$$

$$= \frac{1}{2}||\sum_{i=1}^{n}\alpha_i^* y_i x_i||^2 + \sum_{i=1}^{n}\alpha_i^* - \sum_{i=1}^{n}\alpha_i^* y_i(\sum_{j=1}^{n}\alpha_j^* y_j x_j)^\top\mathbf{x}_i \tag{21}$$

$$= \frac{1}{2}||\sum_{i=1}^{n}\alpha_i^* y_i x_i||^2 + \sum_{i=1}^{n}\alpha_i^* - \sum_{i=1}^{n}(\sum_{j=1}^{n}\alpha_j^* y_j x_j)^\top(\alpha_i^* y_i\mathbf{x}_i) \tag{22}$$

$$= \frac{1}{2}||\sum_{i=1}^{n}\alpha_i^* y_i x_i||^2 + \sum_{i=1}^{n}\alpha_i^* - ||\sum_{i=1}^{n}\alpha_i^* y_i x_i||^2 \tag{23}$$

$$= \sum_{i=1}^{n}\alpha_i^* - \frac{1}{2}||\sum_{i=1}^{n}\alpha_i^* y_i x_i||^2 \tag{24}$$

$$= \mathbf{1}^\top \alpha^* - \frac{1}{2}\alpha^* \mathbf{Q} \alpha^* \tag{25}$$

where we let $\mathbf{Q} = (\operatorname{diag} \mathbf{y})\mathbf{X}\mathbf{X}^T(\operatorname{diag} \mathbf{y})$.

Noting the previous constraints resulting from the KKT conditions, we then get the dual problem is the following maximization.

$$\max_{\alpha} \mathbf{1}^\top \alpha^* - \frac{1}{2}\alpha^* \mathbf{Q} \alpha^* \tag{26}$$

$$s.t. \quad \alpha^\top \mathbf{y} = 0 \tag{27}$$

$$0 \leq \alpha_i \leq C\phi_i \quad i = 1, \ldots, n \tag{28}$$

(c) **From the dual formulation above, how would you kernelize the SVM with custom margins? What role does the ϕ_i play in the kernelized version?**

Solution: We can kernelize the SVM in the same way as in the normal SVM (e.g., note that in the definition of $\mathbf{Q}$, we have an $XX^\top$; we replace this with the kernel matrix K).

The ϕ_i serve to adjust the constraints on α_i. If ϕ_i is very large (in the primal this means we want the margin violations to be small for the data point x_i), the constraint on α_i will be very loose. Similarly, if ϕ_i is very small (in the primal this means we allow the margin violation to be large for the data point x_i), the constraint on α_i becomes much tighter.

3 Imputation of Missing Data using EM

This question is about adapting EM to a discrete problem of missing data.

Recall that in the case of a mixture of Gaussians we did soft imputation of the individual cluster assignments in the E-step and then estimation of $\boldsymbol{\theta}$, the set of parameters defining the individual Gaussians and the prior for Z, the random variable defining the cluster assignment, in the M-step. We iterates this process until convergence. EM, however, can be generalized to many other settings involving hidden variables and parameter estimation. In the mixture of Gaussian case, our hidden variables were the cluster assignments. In the following problem, we will explore how to apply EM to the setting where our hidden variables are missing data instead.

Suppose we have $\mathbf{Y} = [Y_1, Y_2, Y_3, Y_4, Y_5]$ where Y_i are random variables which are jointly distributed multinomially with probabilities $(\frac{1}{2}, \frac{\theta}{4}, \frac{1}{4}(1-\theta), \frac{1}{4}(1-\theta), \frac{\theta}{4})$. Recall that for a multinomial distribution is a generalization of the binomial distribution and with a probability mass function parameterized by event probabilities $p_1, \ldots, p_k$. The PMF is $p(y_1, \ldots, y_k; p_1, \ldots, p_k) = \frac{n!}{y_1! \ldots y_5!} p_1^{y_1} \cdots p_k^{y_k}$.

In this problem, we observe data coming from an experiment that had 8 observations:

$$\mathbf{y} = [y_1, y_2, y_3, y_4, y_5] = [?, ?, 2, 2, 3]$$

where y_1 and y_2 are missing from our observations. Because there were 8 observations taken, we know that $y_{sum} = y_1 + y_2 = 1$ but we don't know which one is 1 and which one is 0.

To be clear, we know what the distribution is, but we don't know the parameter θ in the distribution and we don't have values for the first two categories.

(a) **What is the log likelihood function, i.e. $p(\mathbf{y}|\theta)$?**

Solution:

$$p(\mathbf{y}|\theta) = \frac{n!}{y_1!y_2!y_3!y_4!y_5!}\left(\frac{1}{2}\right)^{y_1}\left(\frac{\theta}{4}\right)^{y_2}\left(\frac{1-\theta}{4}\right)^{y_2}\left(\frac{1-\theta}{4}\right)^{y_3}\left(\frac{\theta}{4}\right)^{y_4}$$

The log-likelihood is then:

$$\ell_\theta(\mathbf{y}) = (y_2 + y_5)\log\theta + (y_3 + y_4)\log(1-\theta)$$

(b) Recall that the EM algorithm iterates between soft imputation for our unobserved variables (E-step) and performing parameter estimation via maximization (M-step). In the E-step for the mixture of Gaussian case, we computed $p_{\boldsymbol{\theta}}(Z_i = k|X = x_i)$ where Z_i is a Bernouilli random variable determining the cluster assignment for x_i. Here our unobserved/hidden variables are Y_1 and Y_2, so we would like to compute $q_k^{t+1} = p_{\boldsymbol{\theta}}(Y_1 = k, Y_2 = \bar{k}|y_{sum})$ for $k = 0, 1$ since we know $y_1 + y_2 = 1$. Here $\bar{k}$ just means the opposite so $\bar{k} = 1$ if $k = 0$ and $\bar{k} = 0$ if $k = 1$. **Derive the q_0^{t+1}, q_1^{t+1} given the current $\boldsymbol{\theta}^t$.**

Solution: More generally, Y_1 can be thought of as a Binomial distribution with sample size 1 and parameter $p = \frac{\frac{\theta}{4}}{\frac{1}{8}+\frac{\theta}{4}} = \frac{2\theta}{1+2\theta}$ and Y_2 has parameter $p = \frac{1}{1+2\theta}$. Thus,

$$\begin{aligned}
q_{1,1}^{t+1} &= p_{\boldsymbol{\theta}}(Y_1 = 0|y_{sum}) = \frac{2\theta}{1+2\theta} \\
q_{1,2}^{t+1} &= p_{\boldsymbol{\theta}}(Y_1 = 1|y_{sum}) = \frac{1}{1+2\theta} \\
q_{2,1}^{t+1} &= p_{\boldsymbol{\theta}}(Y_2 = 0|y_{sum}) = \frac{1}{1+2\theta} \\
q_{2,2}^{t+1} &= p_{\boldsymbol{\theta}}(Y_2 = 1|y_{sum}) = \frac{2\theta}{1+2\theta}
\end{aligned}$$

(c) Recall that in the M-step we update our estimate for the parameters θ by maximizing the expected complete log-likelihood: $\theta^{t+1} = \text{argmax}_\theta \mathbb{E}_q\left[\log\left(p_{\boldsymbol{\theta}}(\mathbf{Y}|y_{sum}, y_3, y_4, y_5)\right)\right]$. **Write the expression for the complete log-likelihood and the closed form expression for the expected complete log-likelihood in terms of $q_0^{t+1}, q_1^{t+1}, y_3, y_4, y_5$, and θ?**

Solution: The complete log-likelihood: $\log\big(p_{\boldsymbol{\theta}}(\mathbf{Y}|y_{sum}, y_3, y_4, y_5)\big) = (Y_2 + y_5)\log\theta + (y_3 + y_4)\log(1-\theta)$.

$$\begin{aligned}
&\mathbb{E}_q\left[\log\big(p_{\boldsymbol{\theta}}(\mathbf{Y}|y_{sum}, y_3, y_4, y_5)\big)\right] \\
&= \mathbb{E}_q\left[(Y_2 + y_5)\log\theta + (y_3 + y_4)\log(1-\theta)|y_{sum}\right] \\
&= (y_2^{t+1} + y_5)\log\theta + (y_3 + y_4)\log(1-\theta)
\end{aligned}$$

(d) **Maximize the expression for the expected complete log-likelihood to obtain an expression for $\boldsymbol{\theta}^{t+1}$.**

Solution: Taking the derivative of the expected complete log-likelihood and setting to 0 gives:

$$\begin{aligned}
(1-\theta)(y_2^{t+1} + y_5) &= \theta(y_3 + y_4) \\
y_2^{t+1} + y_5 &= \theta(y_2^{t+1} + y_3 + y_4 + y_5)
\end{aligned}$$

Solving $\theta^{t+1} = \frac{y_2^{t+1} + y_5}{y_2^{t+1} + y_3 + y_4 + y_5}$.

(e) Using q_0^{t+1}, q_1^{t+1} computed in the E-step, **obtain a reasonable estimate for y_1 and y_2 and justify your answer.**

Solution: One reasonable way would be to take the expectation:

$$\begin{aligned}
\hat{y}_1^{t+1} &= \frac{2\theta^t}{1 + 2\theta^t} \\
\hat{y}_2^{t+1} &= \frac{1}{1 + 2\theta^t}
\end{aligned}$$

Another would be to set the $\hat{y}_1^{t+1} = 1$ if $p_{\boldsymbol{\theta}}(Y_1 = 1|y_{sum}) > p_{\boldsymbol{\theta}}(Y_2 = 1|y_{sum})$ and vice versa.

(f) Let's consider how we might approach the problem using the MLE directly. One way to do this would be to marginalize out Y_1 and Y_2 of the log-likelihood by summing over all possible pairs: $(Y_1 = 0, Y_2 = 1)$ and $(Y_1 = 1, Y_2 = 0)$. **Write the expression for the MLE minimization for θ. Explain how you would compute an estimate for θ, but no need to compute it. How would you go from there to imputing Y_1 and Y_2?**

Solution: You would take derivative of the expression

$$\frac{\theta}{4}\left(\frac{1-\theta}{4}\right)^2\left(\frac{1-\theta}{4}\right)^2\left(\frac{3+2\theta}{8}\right)^3 + \frac{1}{8}\left(\frac{1-\theta}{4}\right)^2\left(\frac{1-\theta}{4}\right)^2\left(\frac{3+2\theta}{8}\right)^3$$

and set it to 0 and solve for θ.

4 Coin tossing with unknown coins (35 points)

This question is about adapting EM and the spirit of k-means to a discrete problem of tossing coins.

We have a bag that contains two kinds of coins that look identical. The first kind has probability of heads p_a and the other kind has probability of heads p_b, but we don't know these. We also don't know how many of each kind of coin are in the bag; so the probability, α_a, of drawing a coin of the first type is also unknown (and since $\alpha_a + \alpha_b = 1$, we do not need to separately estimate α_b, the probability of drawing a coin of the second type).

What we have is n pieces of data: for each data point, someone reached into the bag, pulled out a random coin, tossed it ℓ times and then reported the number h_i which was the number of times it came up heads. The coin was then put back into the bag. This was repeated n times. The resulting n head-counts $(h_1, h_2, h_3, \ldots, h_n)$ constitute our data.

Our goal is to estimate p_a, p_b, α_a from this data in some reasonable way.

For this problem, the binomial distribution can be good to have handy:

$$P(H = h) = \binom{\ell}{h} p^h (1-p)^{\ell - h}$$

for the probability of seeing exactly h heads having tossed a coin ℓ times with each toss independently having probability p of turning up heads. Also recall that the mean and variance of a binomial distribution are given respectively by ℓp and $\ell p(1-p)$.

(a) (10 pts) **How would you adapt the main ideas in the k-means algorithm to construct an analogous approach to estimating $\widehat{p}_a, \widehat{p}_b, \widehat{\alpha}_a$ from this data set? Give an explicit algorithm, although it is fine if it is written just in English.**

Solution: They key ideas of k-means are: (1) introduce some notion of each point belonging to each "cluster"–here the analogue to cluster from k-means is which coin the data came from, either a or b, (2) introduce some parametric way to define each cluster, such as a "centroid"–here one analogue to the centroid vector could be the scalar representing the probability of heads for each coin, p_a and p_b, (3) introduce some notion of distance between each observed data point and each "cluster", such as the Euclidean distance between a data point and a centroid center—here the analogue could be something like $d(h_i, \widehat{p}_a; l) = |\frac{h_i}{l} - \widehat{p}_a|$, (4) an iterative procedure that goes back and forth between assigning each point fully to one cluster (coin) or the other and updating the "centroid" parameters.

Thus, one example of a solution would have been:

(a) Randomly initialize the values $\widehat{p}_a$ and $\widehat{p}_b$ between 0 and 1. (Equivalently could instead have initialized the cluster assignments, in which case reverse the next two steps).

(b) Assign each data point to coin a if $|\frac{h_i}{l} - \widehat{p}_a| >= |\frac{h_i}{l} - \widehat{p}_a|$, and otherwise coin b. Denote this assignment, z_i, where $z_i \in \{a, b\}$.

(c) Update the parameter $\widehat{p}_a$ using $\widehat{p}_a = \frac{1}{|\{i|z_i=a\}|} \sum_{h_i|z_i=a} \frac{h_i}{l}$.

(d) Repeat the two steps above until convergence, where convergence could be that the parameters have stopped moving more than T away from where they were at the last iteration.

(e) Set the parameter α_a using $\alpha_a = \frac{1}{n} \sum_{h_i | z_i = a} 1$. Similarly for α_b.

(b) (8 pts) Suppose that the true $p_a = 0.4$ and the true $p_b = 0.6$ and $\alpha_a = 0.5$, and $\ell = 5$. For $n \to \infty$, **will your "k-means" based estimates (those from the preceding question) for $\widehat{p}_a$ and $\widehat{p}_b$ yield the correct parameter estimates (namely, $\widehat{p}_a = 0.4$ and $\widehat{p}_b = 0.6$)? Why or why not?**

Hint: Draw a sketch of the typical histograms of the number of heads of each coin on the same axes.

Solution: The binomials have means of 2 and 3 for class a and b respectively and variance of 1.2. Thus if one draws out the histograms on the same plot, we see that their distributions overlap by a substantial amount. Because "K-means" does "hard" assignment of each point to each "cluster" (coin), then this overlap will cause problems since it is not clear which coin each data point was generated from. To properly handle this we would need probabilistic assignments, but "k-means" uses hard assignments. Thus, because of this ambiguity, some data which were truly generated from class b will get assigned to class a, and vice-versa, even as the algorithm progresses–it is inherent to the hard assignments. Thus the final estimates for $\widehat{p}_a$ and $\widehat{p}_b$ will be biased (and thus incorrect even with infinite data). In particular, $\widehat{p}_a$ will be too low and $\widehat{p}_b$ will be too high.

(c) (17 pts) How would you adapt the EM for Gaussian Mixture Models that you have seen to construct an EM algorithm for estimating $\widehat{p}_a, \widehat{p}_b, \widehat{\alpha}_a$ from this data set?

You don't have to solve for the parameters in closed form, but (i) **write down the E-step update equations (i.e. write down the distributions that should be computed for the E-step — not in general, but specifically for this problem) and (ii) the objective function that gets maximized for the M-step and also what you are maximizing with respect to (again, not just the general form, but specific to this problem).** If you introduce any notation, be sure to **explain what everything means. Explain in words what the E- and M-steps are doing on an intuitive level.**

Solution:

Intuitively, in English: The E-step fills in the "missing" data (the assignment of data point to the coin it was generated from) in a probabilistic manner. The M-step uses this "probabilistically filled in data" to perform maximum likelihood on the data.

Formally now. First, let $z_i \in \{a, b\}$ represent which class data item i was generated from.

Then, in the E-step, we compute the posterior over this hidden variable for each data point. In particular we compute

$$p(z_i = a|\hat{p}_a, \hat{p}_b, \hat{\alpha}_a l, h_i) = \frac{r_i^a}{r_i^a + r_i^b}$$

where $r_i^a \equiv \alpha_a \text{Binomial}(H = h_i; \hat{p}_a, l)$. Similarly, compute the analogue for class b.

Then, in the M-step, we introduce a short-hand notation, $q_n^a \equiv p(z_i = a|\hat{p}_a, \hat{p}_b, \hat{\alpha}_a l, h_i)$ and similarly for q_n^b. Using this, we then need to maximize the likelihood of each binomial distribution, using the "soft" assignment of data to each binomial:

$$\hat{p}_a = \arg\max_{p_a} \sum_i q_i^a \log \text{Binomial}(H = h_i; \widehat{p}_a, l)$$
$$\hat{p}_a = \arg\max_{p_b} \sum_i q_i^b \log \text{Binomial}(H = h_i; \widehat{p}_b, l)$$
$$\hat{\alpha_a} = \frac{\sum_i q_i^a}{n}$$

Implicitly, it is also true that $\hat{\alpha_b} = 1 - \hat{\alpha_a}$.

One could instead have written down the objective function for $\hat{\alpha}_a$ as requested:

$$\hat{\alpha}_a = \arg\max_{\alpha_a} \sum_n q_n^a \log(\alpha_a) + q_n^a \log \text{Binomial}(H = h_n; p_a, l) + \ldots$$
$$\ldots + q_n^b \log(1 - \alpha_a) + q_n^b \log \text{Binomial}(H = h_n; p_b, l)$$