CS 189 Introduction to Machine Learning
Summer 2018 HW7

1 Getting Started

Read through this page carefully. You may typeset your homework in latex or submit neatly handwritten/scanned solutions. Please start each question on a new page. Deliverables:

1. Submit a PDF of your writeup, **with an appendix for your code**, to assignment on Gradescope, "HW7 Write-Up". If there are graphs, include those graphs in the correct sections. Do not simply reference your appendix.
2. If there is code, submit all code needed to reproduce your results, "HW7 Code".
3. If there is a test set, submit your test set evaluation results, "HW7 Test Set".

After you've submitted your homework, watch out for the self-grade form.

(a) Who else did you work with on this homework? In case of course events, just describe the group. How did you work on this homework? Any comments about the homework?

(b) Please copy the following statement and sign next to it. We just want to make it *extra* clear so that no one inadverdently cheats.

I certify that all solutions are entirely in my words and that I have not looked at another student's solutions. I have credited all external sources in this write up.

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:
Note: Soft-margin SVM is an extension of the standard SVM that allows for violations of the margin constraints by introducing slack variables ξ_i's. You can refer to section 1.3 of course note 20 for more details.

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

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

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

$$\max_{\alpha} \alpha^\top \mathbf{1} - \frac{1}{2}\alpha^\top \mathbf{Q}\alpha \tag{8}$$

$$s.t. \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{9}$$

$$0 \leq \alpha_i \leq C \quad i = 1, \cdots, n \tag{10}$$

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.

(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?**

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 $p(y_1, \ldots, y_k; p_1, \ldots, p_k) = \frac{n!}{y_1! \ldots y_5!} p_1^{y_1} \ldots p_k^{y_k}$.

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. $p(\mathbf{y}|\theta)$?**

(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 $\boldsymbol{\theta}^t$.**

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

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

(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.**

(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?**

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.**

(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.

(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.**

5 Your Own Question

Write your own question, and provide a thorough solution.

Writing your own problems is a very important way to really learn the material. The famous "Bloom's Taxonomy" that lists the levels of learning is: Remember, Understand, Apply, Analyze, Evaluate, and Create. Using what you know to create is the top-level. We rarely ask you any HW questions about the lowest level of straight-up remembering, expecting you to be able to do that yourself. (e.g. make yourself flashcards) But we don't want the same to be true about the highest level.

As a practical matter, having some practice at trying to create problems helps you study for exams much better than simply counting on solving existing practice problems. This is because thinking about how to create an interesting problem forces you to really look at the material from the perspective of those who are going to create the exams.

Besides, this is fun. If you want to make a boring problem, go ahead. That is your prerogative. But it is more fun to really engage with the material, discover something interesting, and then come up with a problem that walks others down a journey that lets them share your discovery. You don't have to achieve this every week. But unless you try every week, it probably won't happen ever.