CS221 Fall 2015 Homework [1]

SUNet ID: [lguan]
Name: [Leying Guam]
Collaborators: [list all the people you worked with]

By turning in this assignment, I agree by the Stanford honor code and declare that all of this is my own work.

Problem 1

(a) the minimizer $x^* = \frac{\sum_{i=1}^n w_i b_i}{\sum_{i=1}^n w_i}$; if some w_i's are negative, the solution to this problem will depend on w_i's value. If $\sum_{i=1}^n w_i > 0$, the original question still holds; otherwise, the problem is not well defined as the minimium value is $-\infty$.

(b) we have $f(x) \leq g(x)$: because $ax_j \leq \max_{a\in\{1,-1\}} x_j, \forall j = 1,..,d, \Rightarrow \sum_{j=1}^d ax_j \leq \sum_{j=1}^d \max_{a\in\{1,-1\}} x_j, \forall x \Rightarrow f(x) \leq g(x)$

(c) The expected total points Y I earn is;

$$
\begin{aligned}
E(Y) & \sum_{k=1}^{\infty}(\sum_{i=0}^{k-1} C_{k-1}^i (\frac{1}{4})^i (\frac{3}{4})^{k-1-i} ir)(\frac{2}{3})^{k-1}\frac{1}{3} \\
&= \frac{1}{4}\sum_{k=1}^{\infty}(\frac{2}{3})^{k-1}\frac{1}{3}(k-1)r \\
&= \frac{1}{4} * (3-1) * r \\
&= \frac{r}{2}
\end{aligned}
$$

(d) By taking derivative of the log-likelihood and let it be zero, we have

$$\frac{2}{p} - \frac{3}{1-p} = 0$$

thus $p = \frac{2}{5}$

(e)

$$
\begin{aligned}
\frac{\partial f(w)}{\partial w_k} &= \sum_{i=1}^n \sum_{j=1}^n (a_i^T w - b_j^T w)(a_{ik} - b_{jk}) + 2\lambda w_k \\
\Rightarrow \nabla f(w) &= \sum_{i=1}^n \sum_{j=1}^n (a_i^T w - b_j^T w)(a - b) + 2\lambda w
\end{aligned}
$$

Problem 2

(a) $O(n^{4+4+4+4+4}) = O(n^20)(c = 20)$ because each component has order $O(n^4)$ posibilities

(b) Use dynamic programing and memorize all smaller indexes' minimum cost. And This will have run time $O(n^2)$ because given previous calculation, it takes $O(n)$ to get the new shortest cost for $n \Rightarrow$ the run time is controlled by $O(1 + 2 + ... + n) = O(n^2)$

(c) Because we will ultimately go right n steps and go down n-1 steps, and the complexity is equivalent to calculated different combinations of n-1 down and n-1 right steps, which is $C_{2(n-1)}^{n-1}$

(d) Write f in the following form

$$
\begin{aligned}
f(w) &= \sum_{i=1}^{n} \sum_{j=1}^{n} w^T (a_i - b_j)(a_i - b_j)^T w + \lambda ||w||_2^2 \\
&= w^T (\sum_{i=1}^{n} \sum_{j=1}^{n} (a_i - b_j)(a_i - b_j)^T) w + \lambda ||w||_2^2
\end{aligned}
$$

Thus we can first calculate $(\sum_{i=1}^{n} \sum_{j=1}^{n} (a_i - b_j)(a_i - b_j)^T)$, and then for each w, we only need $O(d^2)$ to update