Homework 2 Solutions
EM, Mixture Models, PCA, Dualitys

CMU 10-715: Machine Learning (Fall 2015)
http://www.cs.cmu.edu/~bapoczos/Classes/ML10715_2015Fall/
OUT: Oct 5, 2015
DUE: Oct 19, 2015, 10:20 AM

1 An EM algorithm for a Mixture of Bernoullis [Eric; 25 pts]

In this section, you will derive an expectation-maximization (EM) algorithm to cluster black and white images. The inputs $x^{(i)}$ can be thought of as vectors of binary values corresponding to black and white pixel values, and the goal is to cluster the images into groups. You will be using a mixture of Bernoullis model to tackle this problem.

For the sake of brevity, you do not need to substitute in previously derived expression in later problems. For example, beyond question 1.1.1 you may use $P(x^{(i)}|p^{(k)})$ in your answers.

1.1 Mixture of Bernoullis

1. (2pts) Consider a vector of binary random variables, $x \in \{0,1\}^D$. Assume each variable x_d is drawn from a Bernoulli(p_d) distribution, so $P(x_d = 1) = p_d$. Let $p \in (0,1)^D$ be the resulting vector of Bernoulli parameters. Write an expression for $P(x|p)$.

   Solution:

$$P(x|p) = \prod_{d=1}^{D} p_d^{x_d}(1-p_d)^{1-x_d}$$

2. (2pts) Now suppose we have a mixture of K Bernoulli distributions: each vector $x^{(i)}$ is drawn from some vector of Bernoulli random variables with parameters $p^{(k)}$, we will call this Bernoulli($p^{(k)}$). Let $\{p^{(1)}, \ldots, p^{(K)}\} = \mathbf{p}$. Assume a distribution $\pi(k)$ over the selection of which set of Bernoulli parameters $p^{(k)}$ is chosen. Write an expression for $P(x^{(i)}|\mathbf{p}, \pi)$.

   Solution: Let A_k be the event that $x = x^{(i)}$ was drawn from $p^{(k)}$. Then:

$$P(x|\mathbf{p},\pi) = \sum_k P(x, A_k|\mathbf{p},\pi) = \sum_k P(x|A_k,\mathbf{p},\pi)P(A_k|\mathbf{p},\pi) = \sum_k \pi_k P(x|p^{(k)})$$

3. (2pts) Finally, suppose we have inputs $X = \{x^{(i)}\}_{i=1\ldots n}$. Using the above, write an expression for the log likelihood of the data X, $\log P(X|\pi, \mathbf{p})$.

   Solution:

$$\log L(\mathbf{p},\pi) = \sum_{i=1}^{N} \log P(x^{(i)}|\mathbf{p},\pi)$$

1.2 Expectation step

1. (4pts) Now, we introduce the latent variables for the EM algorithm. Let $z^{(i)} \in \{0,1\}^K$ be an indicator vector, such that $z_k^{(i)} = 1$ if $x^{(i)}$ was drawn from a Bernoulli($p^{(k)}$), and 0 otherwise. Let $Z = \{z^{(i)}\}_{i=1\ldots n}$. What is $P(z^{(i)}|\pi)$? What is $P(x^{(i)}|z^{(i)}, \mathbf{p}, \pi)$?

Solution: Let $A_k^{(i)}$ be the event that $x^{(i)}$ was drawn from $p^{(k)}$.

$$P(z^{(i)}|\pi) = \prod_{k=1}^{K} \pi_k^{z_k^{(i)}}$$

$$P(x^{(i)}|z^{(i)}, \mathbf{p}, \pi) = \prod_{k=1}^{K} P(x^{(i)}|z^{(i)}, \mathbf{p}, \pi, A_k^{(i)})^{z_k^{(i)}} = \prod_{k=1}^{K} \left[P(x^{(i)}|p^{(k)})\right]^{z_k^{(i)}}$$

2. (2pts) Using the above two quantities, derive the likelihood of the data and the latent variables, $P(Z, X|\pi, \mathbf{p})$.

   Solution:

$$P(Z, X|\pi, \mathbf{p}) = \prod_{i=1}^{N} P(x^{(i)}, z^{(i)}|\pi, \mathbf{p}) = \prod_{i=1}^{N} P(x^{(i)}, |z^{(i)}, \pi, \mathbf{p})P(z^{(i)}|\pi) = \prod_{i=1}^{N} \left[\prod_{k=1}^{K} \left[P(x^{(i)}|p^{(k)})\right]^{z_k^{(i)}}\right] \left[\prod_{k=1}^{K} \pi_k^{z_k^{(i)}}\right]$$

3. (5pts) Let $\eta(z_k^{(i)}) = E[z_k^{(i)}|x^{(i)}, \pi, \mathbf{p}]$. Show that

$$\eta(z_k^{(i)}) = \frac{\pi_k \prod_{d=1}^{D} (p_d^{(k)})^{x_d^{(i)}} (1 - p_d^{(k)})^{1-x_d^{(i)}}}{\sum_j \pi_j \prod_{d=1}^{D} (p_d^{(j)})^{x_d^{(i)}} (1 - p_d^{(j)})^{1-x_d^{(i)}}}$$

   Let $\tilde{\mathbf{p}}, \tilde{\pi}$ be the new parameters that we'd like to maximize, so $\mathbf{p}, \pi$ are from the previous iteration. Use this to derive the following final expression for the E step in the expectation-maximization algorithm:

$$E[\log P(Z, X|\tilde{\mathbf{p}}, \tilde{\pi})|X, \mathbf{p}, \pi] = \sum_{i=1}^{N} \sum_{k=1}^{K} \eta(z_k^{(i)}) \left[\log \tilde{\pi}_k + \sum_{d=1}^{D} \left(x_d^{(i)} \log \tilde{p}_d^{(k)} + (1 - x_d^{(i)}) \log(1 - \tilde{p}_d^{(k)})\right)\right]$$

   Solution:

$$\begin{aligned}
\eta(z_k^{(i)}) &= E[z_k^{(i)}|x^{(i)}, \pi, \mathbf{p}] \\
&= P[z_k^{(i)} = 1|x^{(i)}, \pi, \mathbf{p}] \\
&= \frac{P(x^{(i)}|z_k^{(i)} = 1, \pi, \mathbf{p})P(z_k^{(i)} = 1|\pi, \mathbf{p})}{\sum_{k'} P(x^{(i)}|z_{k'}^{(i)} = 1, \pi, \mathbf{p})P(z_{k'}^{(i)} = 1|\pi, \mathbf{p})} \\
&= \frac{\pi_k \prod_{d=1}^{D} (p_d^{(k)})^{x_d^{(i)}} (1 - p_d^{(k)})^{1-x_d^{(i)}}}{\sum_{k'} \pi_{k'} \prod_{d=1}^{D} (p_d^{(k')})^{x_d^{(i)}} (1 - p_d^{(k')})^{1-x_d^{(i)}}}
\end{aligned}$$

   Next we compute the log likelihood:

$$\begin{aligned}
\log P(Z, D|\pi, \mathbf{p}) &= \sum_{i=1}^{N} \left[\sum_{k=1}^{K} z_k^{(i)} \log \left[P(x^{(i)}|p^{(k)})\right]\right] + \left[\sum_{k=1}^{K} z_k^{(i)} \log \pi_k\right] \\
&= \sum_{i=1}^{N} \sum_{k=1}^{K} z_k^{(i)} \left[\log P(x^{(i)}|p^{(k)}) + \log \pi_k\right] \\
&= \sum_{i=1}^{N} \sum_{k=1}^{K} z_k^{(i)} \left[\log \pi_k + \log \prod_{d=1}^{D} (p_d^{(k)})^{x_d^{(i)}} (1 - p_d^{(k)})^{1-x_d^{(i)}}\right] \\
&= \sum_{i=1}^{N} \sum_{k=1}^{K} z_k^{(i)} \left[\log \pi_k + \sum_{d=1}^{D} \left(x_d^{(i)} \log(p_d^{(k)}) + (1 - x_d^{(i)}) \log(1 - p_d^{(k)})\right)\right]
\end{aligned}$$

   Taking the expected value and replacing $E[z_k^{(i)}] = \eta(z_k^{(i)})$ finishes the solution.

1.3 Maximization step

1. (4pts) We need to maximize the above expression with respect to $\tilde{\pi}, \tilde{\mathbf{p}}$. First, show that the value of $\tilde{\mathbf{p}}$ that maximizes the E step is

$$\tilde{p}^{(k)} = \frac{\sum_{i=1}^N \eta(z_k^{(i)}) x^{(i)}}{N_k}$$

where $N_k = \sum_{i=1}^N \eta(z_k^{(i)})$.

Solution: Setting the derivative to 0:

$$\frac{d}{dp_d^{(k)}} E[\log P(Z, D|\pi, \mathbf{p})] = \sum_{i=1}^N \eta(z_k^{(i)}) \left[\frac{x_d^{(i)}}{p_d^{(k)}} + \frac{1 - x_d^{(i)}}{1 - p_d^{(k)}}\right] = 0$$

Multiply by the denominators:

$$\sum_{i=1}^N \eta(z_k^{(i)}) \left[x_d^{(i)}(1 - p_d^{(k)}) + (1 - x_d^{(i)}) p_d^{(k)}\right] = \sum_{i=1}^N \eta(z_k^{(i)}) \left[-p_d^{(k)} + x_d^{(i)}\right] = 0$$

Solving for $p_d^{(k)}$ results in

$$p_d^{(k)} = \frac{\sum_{i=1}^N \eta(z_k^{(i)}) x_d^{(i)}}{\sum_{i=1}^N \eta(z_k^{(i)})} = \frac{\sum_{i=1}^N \eta(z_k^{(i)}) x_d^{(i)}}{N_k}$$

2. (4pts) Show that the value of $\tilde{\pi}$ that maximizes the E step is

$$\tilde{\pi}_k = \frac{N_k}{\sum_{k'} N_{k'}}$$

The exponential families notation may be useful. Alternatively, you can use Lagrange multipliers.

Solution: We only need to minimize $\sum_{i=1}^N \sum_{k=1}^K \eta(z_k^{(i)}) \log \pi_k$ since the rest is not a function of π. In order to keep π a distribution, we require $\sum_k \pi_k = 1$. Let λ be the dual variable for this constraint:

$$L(\pi, \lambda) = -\sum_{i=1}^N \sum_{k=1}^K \eta(z_k^{(i)}) \log \pi_k + \lambda \left(\sum_{k=1}^K \pi_k - 1\right)$$

Taking the derivate w.r.t. π_k:

$$\frac{d}{d\pi_k} L(\pi, \lambda) = -\sum_{i=1}^N \frac{\eta(z_k^{(i)})}{\pi_k} + \lambda = 0$$

solve for π_k and we get

$$\pi_k = \frac{\sum_{i=1}^N \eta(z_k^{(i)})}{\lambda} = \frac{N_k}{\lambda}$$

Now we solve for λ:

$$L(\lambda) = -\sum_{i=1}^N \sum_{k=1}^K \eta(z_k^{(i)})(\log N_k - \log \lambda) + \left(\sum_{k=1}^K N_k - \lambda\right)$$

Take the derivative w.r.t. λ:

$$\frac{1}{\lambda} \sum_{i=1}^N \sum_{k=1}^K \eta(z_k^{(i)}) - 1 = 0$$

Solve for λ

$$\lambda = \sum_{i=1}^N \sum_{k=1}^K \eta(z_k^{(i)}) = \sum_{k=1}^K N_k$$

2 Clustering Images of Numbers [Eric; 25 pts]

In this section you will use the above algorithm to cluster images of numbers. You will be using the MNIST dataset. Each input is a binary number corresponding to black and white pixels, and is a flattened version of the 28x28 pixel image.

We will use the following conventions:

- N is the number of datapoints, D is the dimension of each input, and K is the number of clusters.
- Xs is an $N \times D$ matrix of the input data, where row i is the pixel data for picture i.
- p is a $K \times D$ matrix of Bernoulli parameters, where row k is the vector of parameters for the kth mixture of Bernoullis.
- mix_p is a $K \times 1$ vector containing the distribution over the various mixtures.
- eta is a $N \times K$ matrix containing the results of the E step, so eta[i,k] $= \eta(z_k^{(i)})$.
- clusters is an $N \times 1$ vector containing the final cluster labels of the input data. Each label is a number from 1 to K.

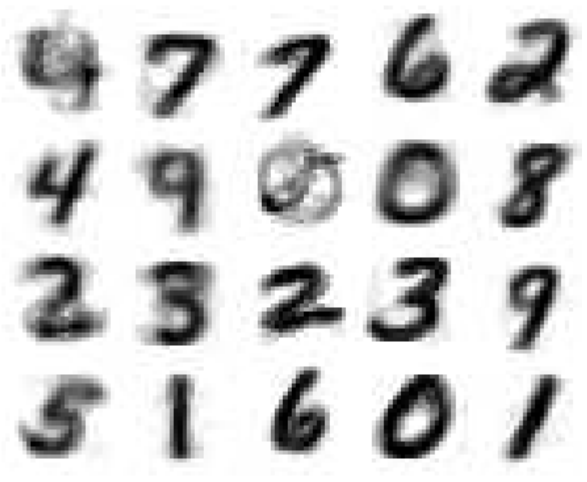
2.1 Programming (20pts)

1. Implement the E step of the algorithm within [eta] = Estep(Xs,p,mix_p), saving your calculated values within eta.
2. Implement the M step of the algorithm within [p,mix_p] = Mstep(Xs, model, alpha1, alpha2). p and mix_p returned by this function should contain the new values that maximize the E step. alpha1, alpha2 are Dirichlet smoothing parameters (explained below).
3. Implement [clusters] = MoBlabels(Xs,p,mix_p). This function will take in the estimated parameters and return the resulting labels that cluster the data.
4. Some hints and tips:
   - The autograder will separately grade the accuracy of your implementation separately for the E-step (4pts), M-step (4pts), and M-step with smoothing (2pts). Finally, it will run your implementation for several iterations on a subset of the MNIST dataset (8pts). A small subset of the MNIST dataset is provided for your convenience.
   - Use the log operator to make your calculations more numerically stable. In particular, pay attention to the calculation of $\eta(z_k^{(i)})$.
   - You will need to avoid zeros in π and $\mathbf{p}$ or else you will take $\log(0) = -\infty$. Use Dirichlet prior smoothing with the parameters α_1, α_2 when updating these variables:

$$\tilde{p}^{(k)} = \frac{\sum_{i=1}^{N} \eta(z_k^{(i)}) x^{(i)} + \alpha_1}{N_k + \alpha_1 D}$$

$$\tilde{\pi}_k = \frac{N_k + \alpha_2}{\sum_{k'} N_{k'} + \alpha_2 K}$$

   - Initialize your parameters $\mathbf{p}$ by randomly sampling from a Uniform$(0,1)$ distribution and normalizing each $p^{(k)}$ to have unit length, and $\pi_k = 1/k$.
   - Running your implementation on the MNIST dataset with $K = 20$ clusters and $\alpha_1 = \alpha_2 = 10^{-8}$ for 20 iterations should give you some results similar to the following (our reference code runs in approximately 10 seconds):



2.2 Analysis

1. (5pts) For each cluster, reshape the pixels into a 28x28 matrix and print the resulting grayscale images. What do you see? Explain, and include one such image in your writeup. See https://www.gnu.org/software/octave/doc/interpreter/Representing-Images.html for help on printing the matrix, or use the provided helper function show_clusters(p, a, b) which will print the mixtures in p in an $a \times b$ grid.

2. (2pts) Using your implemented MoBlabels function, cluster the data. Using the true labels given in yTrain, how many unique digits does each cluster typically have? Are there any clusters that picked out exactly one digit?

3 Kernel PCA [Fish; 15 pts]

Principal component analysis (PCA) is used to emphasize variation and is often used to make data easy to explore and visualize. Suppose we have data $x_1, x_2, \cdots x_N \in \mathbb{R}^d$ that has zero mean. PCA aims at finding a new coordinate system such that the greatest variance by some projection of the data comes to lie on the first coordinate, the second greatest variance on the second coordinate, and so on. The basis vectors of the new coordinate system are the eigenvectors of the co-variance matrix, i.e.,

$$\frac{1}{N}\sum_{i=1}^{N} x_i x_i^T = \sum_{i=1}^{d} \lambda_i v_i v_i^T, \tag{1}$$

where $v_1, v_2, \cdots, v_d$ are orthogonal ($< v_i, v_j >= 0$ for $i \neq j$). Then we can plot our data points on our new coordinate system. The position of each data points in the new coordinate system can be derived by projecting x on to the basis of the new coordinate system, i.e., $v_1, v_2, \cdots v_d$.

3.1 kernel PCA

Often we will want to make linear or non-linear transformation on the data so that they are projected to a higher dimensional space. Suppose there is a transformation function $\phi : \mathbb{R}^d \to \mathbb{R}^l$. We map all the data to a new space through this function, and we have $\phi(x_1), \phi(x_2), \cdots, \phi(x_N)$. We again want to do PCA on the transformed data in the new space. How can we do this?

1. (2pts) Write out the co-variance matrix, C, for $\phi(x_1), \phi(x_2), \cdots, \phi(x_N)$. (Define $\overline{\phi(x)} = \frac{1}{N}\sum_{j=1}^{N} \phi(x_j)$)

Solution:

$$C = \frac{1}{N} \sum_{i=1}^{K} \left(\phi(x_i) - \overline{\phi(x)}\right) \left(\phi(x_i) - \overline{\phi(x)}\right)^T \tag{2}$$

2. (2pts) Now we want to find the basis vectors for the orthogonalized new space by solving eigenvectors of C. Using the definition of an eigenvector, $\lambda v = Cv$, explain why v is a linear combination of $\left(\phi(x_1) - \overline{\phi(x)}\right), \left(\phi(x_2) - \overline{\phi(x)}\right), \cdots, \left(\phi(x_N) - \overline{\phi(x)}\right)$. Since v is a linear combination of these vectors, $v = \sum_{i=1}^N \alpha_i \left(\phi(x_i) - \overline{\phi(x)}\right)$.

   Solution:

$$\lambda v = \frac{1}{N} \sum_{i=1}^{K} \left(\phi(x_i) - \overline{\phi(x)}\right) \left(\phi(x_i) - \overline{\phi(x)}\right)^T v \tag{3}$$

$$v = \frac{1}{\lambda N} \sum_{i=1}^{K} \left(\phi(x_i) - \overline{\phi(x)}\right) \left\langle \phi(x_i) - \overline{\phi(x)}, v \right\rangle \tag{4}$$

$$v = \sum_{i=1}^{N} \alpha_i \left(\phi(x_i) - \overline{\phi(x)}\right). \tag{5}$$

3. (2pts) Before starting to derive α and find out v, we introduce kernel function here. A kernel function is in the form of $k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$. Notice that we do not need to know the function ϕ to calculate $k(x_i, x_j)$. We can simply define a function that is related to x_i, x_j and $k(x_i, x_j) = k(x_j, x_i)$. A classic example is Radial basis function (RBF) kernel, which is $k(x_i, x_j) = \exp(-||x_i - x_j||^2/2\sigma^2)$. In order to use these kernel function, we need to get rid of all the $\phi(x_i)$ and replace them with the kernel function. A kernel matrix is defined as $K \in \mathbb{R}^{N \times N}$ and $K_{ij} = k(x_i, x_j)$. Since we are dealing with non-centered data, we first use a modified kernel matrix $\tilde{K}_{ij} = \left\langle \left(\phi(x_i) - \overline{\phi(x)}\right), \left(\phi(x_j) - \overline{\phi(x)}\right) \right\rangle$. By using the results in question 3.1.1 and 3.1.2 and the definition of eigenvectors, show that

$$N\lambda \tilde{K}\alpha = \tilde{K}^2\alpha. \tag{6}$$

   Solution: Define a matrix $\Phi = [\phi(x_1) - \overline{\phi(x)}, \phi(x_2) - \overline{\phi(x)}, \cdots, \phi(x_N) - \overline{\phi(x)}]$. So we have $v = \Phi\alpha$, $C = \frac{1}{N}\Phi\Phi^T$ and $\tilde{K} = \Phi^T\Phi$. Rewrite $\lambda v = Cv$ with these notations, we have

$$\lambda \Phi\alpha = \frac{1}{N}\Phi\Phi^T\Phi\alpha \tag{7}$$

   Multiply both side by Φ^T, we have

$$\lambda \Phi^T\Phi\alpha = \frac{1}{N}\Phi^T\Phi\Phi^T\Phi\alpha \tag{8}$$

$$N\lambda\tilde{K}\alpha = \tilde{K}^2\alpha \tag{9}$$

4. (2pts) Show that the solutions α in

$$N\lambda\alpha = \tilde{K}\alpha. \tag{10}$$

   are also solutions for equation 6.

   Solution: If α satisfies (10), then it will also satisfies (6).

5. (3pts) Show that

$$\tilde{K} = (I - ee^T)K(I - ee^T). \tag{11}$$

Thus, by solving the eigenvectors of $\tilde{K}$, we get α.

Solution: Define $\Theta = [\phi(x_1), \phi(x_2), \cdots, \phi(x_N)]$. Rewrite $\tilde{K}$ and have

$$\tilde{K} = \left(\Theta - \frac{1}{N}\Theta 11^T\right)^T \left(\Theta - \frac{1}{N}\Theta 11^T\right) \tag{12}$$

$$= \Theta^T\Theta - \frac{1}{N}11^T\Theta^T\Theta - \frac{1}{N}\Theta^T\Theta 11^T + \frac{1}{N^2}11^T\Theta^T\Theta 11^T \tag{13}$$

$$= K - ee^TK - Kee^T + ee^TKee^T \tag{14}$$

$$= (I - ee^T)K(I - ee^T) \tag{15}$$

6. (2pts)After obtaining α, we get the un-normalized basis vector v. To normalize it, what is the factor you need to multiply to v ?

   Solution:

   $$\|v\|_2 = \sqrt{v^Tv} = \sqrt{\alpha^T\Phi^T\Phi\alpha} = \sqrt{\alpha^T\tilde{K}\alpha}. \tag{16}$$

   We need to normalize v by this factor. You shuold not simply write divide by $\sqrt{v}$, because you do not know what is ϕ, and again you need to obtain everything by going through the kernel functions.

7. (2pts) For a data point x, what is its position in the normalized new space? Explain why you can get the new coordinate without explicitly calculate $\phi(x)$.

   Solution: Projection on to the a basis vector $\Phi\alpha$ is

   $$\frac{1}{\sqrt{\alpha^T\tilde{K}\alpha}}\left(\phi(x) - \overline{\phi(x)}\right)\Phi\alpha = \frac{1}{\sqrt{\alpha^T\tilde{K}\alpha}}\sum_{i=1}^{N}\alpha_i\tilde{K}(x, x_i), \tag{17}$$

   so only kernel function is needed. (You can also expend out $\tilde{K}$ and calculate it using K.)

4 Huber function and its application in solving dual problem [Fish; 35pts]

4.1 Huber function

Define a function

$$B(x, d) = \min_{\lambda \geq 0} \lambda + \frac{x^2}{\lambda + d}. \tag{18}$$

where $d > 0$.

1. (3pts) Show that the function

   $$B(x, d) = \begin{cases} \frac{x^2}{d} & \text{if } |x| \leq d \\ 2|x| - d & \text{if } |x| > d \end{cases} \tag{19}$$

   with the minimizer $\lambda^* = \max(0, |x| - d)$. B is often called a huber function.

   Solution: Differentiate $B(x, d)$ with λ, we have $1 - \frac{x^2}{(\gamma+d)^2} = 0$. Solving the function and we get $\lambda = |x| - d$. To match the constraint that $\lambda \geq 0$, we need to have $|x| \geq d$. Besides, with $\lambda = |x| - d$, we have $B(x, d) = |x| - d + \frac{x^2}{|x|} = 2|x| - d$. On the other side, if $|x| < d$, notice that $g(\lambda) = (\lambda + d) + \frac{x^2}{\lambda+d}$ is an increasing function since $g'(\lambda) = 1 - \frac{x^2}{(\lambda+d)^2} \geq 1 - \frac{x^2}{d^2} > 0$, which means you can obtain the minimum value at $\lambda = 0$. So we have $\min_{\lambda\geq 0}\left(\lambda + d + \frac{x^2}{(\lambda+d)} - d\right) = d + \frac{x^2}{d} - d = \frac{x^2}{d}$ at $\lambda = 0$.

2. (3pts) Is the function B convex in x? (Assume d is a fixed constant)

   Solution: At the range of $|x| < d$, $\frac{\partial B(x,d)}{\partial x} = \frac{2x}{d}$ and $\frac{\partial^2 B(x,d)}{\partial x^2} = \frac{2}{d} \geq 0$, so the function is convex in this range. Next we examine the part when $|x| > d$, $\frac{\partial B(x,d)}{\partial x} = 2\text{sign}(x)$ and $\frac{\partial^2 B(x,d)}{\partial x^2} = 0$, so it is convex. Then we examine the 2 points where $|x| = d$, you can see that the differentiation approching from the 2 directions (from $|x| < d$ and $|x| > d$) at this point are both $2\text{sign}(x)$, so the differentiation exists and the twice differentiation is again 0. So the function is also convex at this point. We thus conclude that the function is convex by showing that twice differentation is greater and equal to zero at all points.

3. (3pts) Derive the gradient of function B at any given point (x, d). (Remember $d > 0$.)

   Solution: $x \geq d$, $\partial_{x,d}B(x,d) = (2, -1)$, $x \leq -d$, $\partial_{x,d}B(x,d) = (-2, -1)$, $-d \leq x \leq x$, $\partial_{x,d}B(x,d) = (\frac{2x}{d}, -\frac{x^2}{d^2})$.

4. (3pts) Function B is often used as a penalty term in classification or regression problems, which have the form

$$\min_x L(x) + B(x, d), \tag{20}$$

   where L is the loss function, and d is a parameter with positive value. Describe how parameter d affects the penalty term B on the solution.

   Solution:

   When d is large, than it acts like $l2$ penalty. When d is small, then it acts as $l1$ penalty.

4.2 An application of using Huber loss to solve dual problem

Consider the optimization problem

$$\min_x \sum_{i=1}^{N} (\frac{1}{2} d_i x_i^2 + r_i x_i) \tag{21}$$

$$\text{s.t. } a^T x = 1, x_i \in [-1, 1] \text{ for } i = 1, 2, \cdots n. \tag{22}$$

1. (3pts) Show that the problem has strictly feasible solutions if and only if $||a||_1 > 1$.

   Solution: By Cauchy-schwarz inequality, we have

$$1 = a^T x \leq \|a\|_1 \|x\|_\infty, \|x\|_\infty < 1, \|a\| > 1. \tag{23}$$

   For the other direction, if $\|a\|_1 > 1$, you can set $x_i = \frac{\text{sign}(a_i)}{\|a\|_1}$ and this is a feasible solution to the question.

2. (3pts) Re-express $x_i \in [-1, 1]$ as $x_i^2 \leq 1$, for $i = 1, 2, \cdots, n$. Write down the dual of this problem.

   Solution: The langrange function for this question

$$L(x, u, v) = \sum_{i=1}^{N} (\frac{1}{2} d_i x_i^2 + r_i x_i) + u(a^T x - 1) + \sum_{i=1}^{N} v_i (x_i^2 - 1). \tag{24}$$

   The dual problem is

$$\sup_{u,v} \inf xL(x, u, v). \tag{25}$$

   Now we solve x so that we can have a function that only contains u, v.

$$\frac{\partial L(x, u, v)}{\partial x_i} = d_i x_i + r_i + u a_i + 2 v_i x_i = 0, \tag{26}$$

$$x_i = -\frac{r_i + u a_i}{d_i + 2 v_i}. \tag{27}$$

plug this into the Lagrange function and we have

$$\sum_{i=1}^{N}\left(\frac{1}{2}d_i\frac{(r_i+ua_i)^2}{(d_i+2v_i)^2}-r_i\frac{r_i+ua_i}{d_i+2v_i}-ua_i\frac{r_i+ua_i}{d_i+2v_i}+v_i\left(\frac{(r_i+ua_i)^2}{(d_i+2v_i)^2}-1\right)\right)-u \tag{28}$$

$$=\sum_{i=1}^{N}\left(-\frac{\frac{1}{2}(r_i+ua_i)^2}{2v_i+d_i}-v_i\right) \tag{29}$$

So the dual problem is

$$\max_{u,v}\sum_{i=1}^{N}\left(-\frac{\frac{1}{2}(r_i+ua_i)^2}{2v_i+d_i}-v_i\right)-u \tag{30}$$

$$\text{s.t.} v_i \geq 0 \text{ for } i \in [N] \tag{31}$$

Or you can write it as

$$\min_{u,v} u+\frac{1}{2}\sum_{i=1}^{N}\left[\frac{(r_i+ua_i)^2}{2v_i+d_i}+2v_i\right] \tag{32}$$

$$\text{s.t.} v_i \geq 0 \text{ for } i \in [N] \tag{33}$$

3. (6pts) Write down the KKT condition for this problem. Does it characterize the optimal solution?

   Solution: Yes, KKT characterize the optimal solution. # Stationary:
   $\nabla_x L(x,u,v)=0$
   # Primal feasible:
   $x_i^2 \leq 1$
   $a^T x = 1$
   # Dual feasible:
   $v_i \geq 0$, for $i \in [N]$
   Complementary Slackness:
   $v_i(x_i^2-1)=0$ for $i \in [N]$.

4. (5pts) Show that we can further reduce the dual problem to a one-dimensional convex problem

$$\min_{\mu} \mu+\frac{1}{2}\sum_{i=1}^{N} B(r_i+\mu a_i, d_i), \tag{34}$$

   where B is the Huber function defined in the previous section.

   Solution: Look at (30), it is exactly in this form, where $2v_i$ is the λ, and (r_i+ua_i) is x.

5. (3pts) Describe an algorithm to solve this dual problem. What is the time complexity for your algorithm? We have shown that Huber function is convex and derive its gradient in the question 4.1.3, so we can simply apply gradient descent to solve it. The time complexity is $O(Nk)$, where k is the number of step.

6. (3pts) How can you recover an optimal primal solution x after solving the dual?

   Solution: After deriving u, v, we can recover x by

$$x_i=-\frac{r_i+ua_i}{d_i+2v_i} \tag{35}$$

   as stated in (25).