Feel free to work with other students, but make sure you write up the homework and code on your own (no copying homework *or* code; no pair programming). Feel free to ask students or instructors for help debugging code or whatever else, though.

The starter files for problem 2 can be found under the Resource tab on course website. The plot for problem 2 generated by the sample solution has been included in the starter files for reference. Please print out all the graphs generated by your own code and submit them together with the written part, and make sure you upload the code to your Github repository.

1 (Murphy 11.3 - EM for Mixtures of Bernoullis) Show that the M step for ML estimation of a mixture of Bernoullis is given by

$$\mu_{kj} = \frac{\sum_i r_{ik} x_{ij}}{\sum_i r_{ik}}.$$

Show that the M step for MAP estimation of a mixture of Bernoullis with a $\beta(a, b)$ prior is given by

$$\mu_{kj} = \frac{(\sum_i r_{ik} x_{ij}) + a - 1}{(\sum_i r_{ik}) + a + b - 2}.$$

We have the complete data log likelihood

$$\ell(\mu) = \sum_i \sum_k r_{ik} \log P(x_i \mid \theta_k) = \sum_i \sum_k r_{ik} \sum_j x_{ij} \log \mu_{kj} + (1 - x_{ij}) \log (1 - \mu_{kj})$$

where i is the datapoint index, k is the component, and j is the dimension index of the D dimensional bit vectors. Taking the derivative with respect to μ_{kj}, we have

$$\frac{\partial \ell}{\partial \mu_{kj}} = \sum_i r_{ik} \left(\frac{x_{ij}}{\mu_{kj}} - \frac{1 - x_{ij}}{1 - \mu_{kj}} \right) = \sum_i r_{ik} \left(\frac{x_{ij} - \mu_{kj}}{\mu_{kj}(1 - \mu_{kj})} \right) = \frac{\sum_i r_{ik} (x_{ij} - \mu_{kj})}{\mu_{kj}(1 - \mu_{kj})} = 0.$$

This gives the optimality condition $\sum_i r_{ik} x_{ij} = \mu_{kj} \sum_i r_{ik}$, which gives the desired result.

■

We have the complete data log likelihood plus the log prior (ignoring the π term as we are maximizing without regard to them)

$$\ell(\mu) = \sum_i \sum_k r_{ik} \log P(x_i|\mu_k) + \log P(\mu_k)$$

$$= \sum_i \sum_k r_{ik} \Big(\sum_j x_{ij} \log \mu_{kj} + (1-x_{ij}) \log (1-\mu_{kj}) \Big) + (a-1) \log \mu_{kj} + (b-1) \log (1-\mu_{kj}).$$

Taking derivatives, we have

$$\frac{\partial \ell}{\partial \mu} = \sum_i \left(\frac{r_{ik} x_{ij} + a - 1}{\mu_{kj}} - \frac{r_{ik}(1-x_{ij}) + b - 1}{1-\mu_{kj}} \right)$$

$$= \frac{1}{\mu_{kj}(1-\mu_{kj})} \sum_i r_{ik} x_{ij} - r_{ik}\mu_{kj} + a - 1 - \mu_{kj} a + \mu_{kj} - \mu_{kj} b + \mu_{kj}$$

$$= \frac{1}{\mu_{kj}(1-\mu_{kj})} \left[\sum_i r_{ik} x_{ij} - \mu_{kj} \Big(\sum_i r_{ik} + a + b - 2 \Big) + a - 1 \right] = 0.$$

This gives the optimality condition $\sum_i r_{ik} x_{ij} + a - 1 = \left(\sum_i r_{ik} + a + b - 2 \right) \mu_{kj}$, which gives the desired result. Note that if $a = b = 1$, we arrive at the original maximum likelihood estimate, which makes sense since $\beta(1,1)$ is a uniform distribution over $[0,1]$ so it is as if their is no prior at all.

2 (Lasso Feature Selection) In this problem, we will use the online news popularity dataset we used in hw2pr3. In the starter code, we have already parsed the data for you. However, you might need internet connection to access the data and therefore successfully run the starter code.

First, ignoring undifferentiability at $x = 0$, take $\frac{\partial |x|}{\partial x} = \text{sign}(x)$. Using this, show that $\nabla \|\mathbf{x}\|_1 = \text{sign}(\mathbf{x})$ where sign is applied elementwise. Derive the gradient of the ℓ_1 regularized linear regression objective

$$\text{minimize: } \|A\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_1$$

Then, implement a gradient descent based solution of the above optimization problem for this data. Produce the convergence plot (objective vs. iterations) for a non-trivial value of λ. In the same figure (and different axes) produce a 'regularization path' plot. Detailed more in section 13.3.4 of Murphy, a regularization path is a plot of the optimal weight on the y axis at a given regularization strength λ on the x axis. Armed with this plot, provide an ordered list of the top five features in predicting the log-shares of a news article from this dataset (with justification).

First, let's wrap the gradient descent step with a thredhold function

$$\text{prox}_\gamma(x)_i = \begin{cases} x_i - \gamma & x_i > \gamma \\ 0 & |x_i| \le \gamma \\ x_i + \gamma & x_i < -\gamma \end{cases}$$

so that each iterate $x_{i+1} = \text{prox}_\gamma(x_i - \gamma \nabla f(x_i))$ where γ is the learning rate.

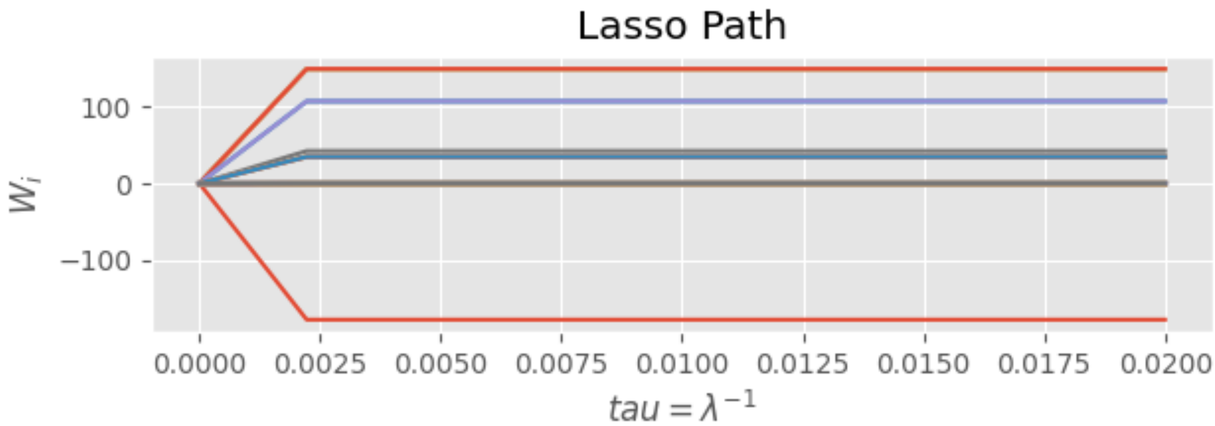
Thus, $\frac{\partial \|x\|_1}{\partial x_i} = \frac{\partial \sum |x_i|}{\partial x_i} = \text{sign}(x_i)$. It follows that $\nabla \|x\|_1 = \text{sign}(x_i)$.

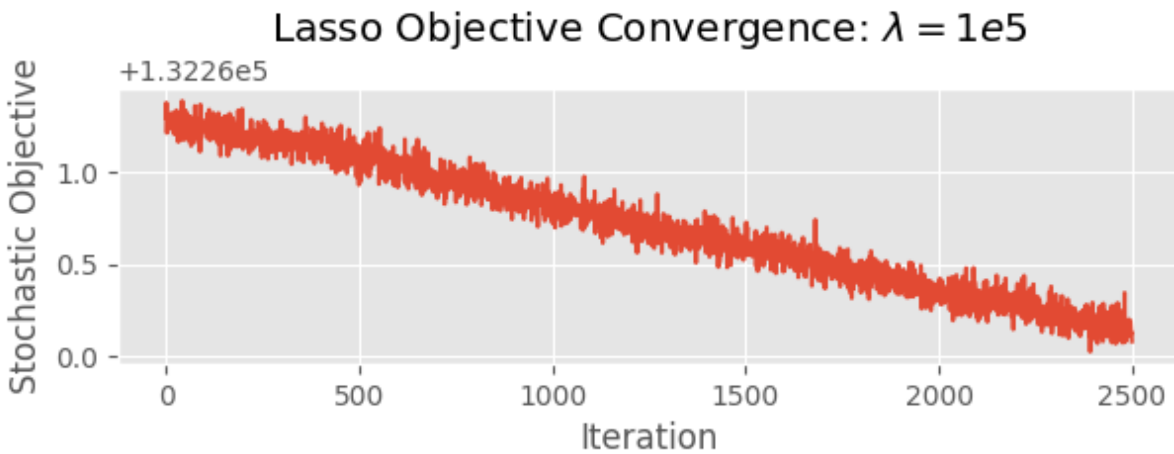
We can then see

$$\nabla \|Ax - b\|_2^2 + \lambda \|x\|_1 = \nabla x^T A^T A x - 2b^T A x + b^T b + \lambda \|x\|_1$$
$$= 2A^T A x - 2b^T A + \lambda\, \text{sign}(x).$$

The plot is attached below. Since we instantiated our weights with the least squares estimate, we can see our lasso objective not moving significantly, although if we look at sparsity over time it does in fact increase.

■


Lasso Path
W_i
100
0
−100
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150
0.0175
0.0200
$tau = \lambda^{-1}$


Lasso Objective Convergence: $\lambda = 1e5$
+1.3226e5
Stochastic Objective
1.0
0.5
0.0
0
500
1000
1500
2000
2500
Iteration