Statistical Machine Learning – Homework 5 Solution

Credit to Shuaiwen Wang

Problem 1

Proof. We compute the distance matrix between the samples:

dis.	A	B	C	D	E	F
A		1	2	6	4	6
B			3	5	3	5
C				4	2	4
D					2	4
E						2
F						

The smallest distance is that between A and B, so we form a new cluster {A,B}. Then, we recompute the distance matrix between our current set of clusters:

dis.	{A, B}	C	D	E	F
{A, B}		2	6	4	6
C			4	2	4
D				2	4
E					2
F					

At this second step, we link all the clusters, because every cluster is within a distance 2 to another cluster. We obtain the single cluster {A, B, C, D, E, F}. □

Problem 2

Proof. There are two bigram models, one describing the word distribution of spam, one of email. Each bigram is a family of multinomial distributions,

$$\{P_{\text{spam}}(w_i|w_{i-1}) \text{ where } w_{i-1} \in \text{vocabulary}\} \quad \text{and} \quad \{P_{\text{email}}(w_i|w_{i-1}) \text{ where } w_{i-1} \in \text{vocabulary}\} \ .$$

The classifier is trained by estimating using maximum likelihood estimation:

- Each distribution $P_{\text{spam}}(w_i|w_{i-1})$ is estimated by MLE from all pairs of words (w, w') in the spam data set with $w_{i-1} = w'$.
- The distributions $P_{\text{email}}(w_i|w_{i-1})$ are estimated similarly, but from the email data set.
- We also need to estimate $\pi(\text{spam})$ and $\pi(\text{email})$. This is estimated by the relative size of $\mathcal{X}_s$ and $\mathcal{X}_e$.

To classify a text of length M, we compute

$$\text{probability of spam given the text} \propto \pi(\text{spam}) \prod_{i=2}^{M} P_{\text{spam}}(w_i|w_{i-1})$$

and

$$\text{probability of email given the text} \propto \pi(\text{email}) \prod_{i=2}^{M} P_{\text{email}}(w_i|w_{i-1}) \ .$$

We then classify according to which of the two probabilities is higher. □

Problem 3

Proof. Please check the code for details. Here I attach the figure with $K = 3, 4, 5$. $\tau = 0.01$ through out the simulation. Here are several remarks for the coding part:

- To initialize t, we will sample K row vectors from matrix H. Then it is required to normalize the vector t. Notice that here t is the parameter of the multinomial distribution, the normalization means to set `t = t / sum(t)` such that the summation of the entries of each t is 1;
- Obviously, we also need to initialize the mixing probability c. A natural way is to set `c = rep(1 / K, K)`;

- As mentioned by the homework description, we can add each components of t by a small quantity to avoid numerical issue from taking `log(t)`. While there is another issue here. Since entries of t is and will always be very small, `log(t)` will give negative entries, which will become quite negative after multiplied by H (around -80 to -700). This will bring problem when you calculate exponential of these numbers and then normalize them. A method to solve this is: suppose you have `vec = c(a, b, c)` with quite negative a, b, c, and you want to calculate $\frac{e^a}{e^a+e^b+e^c}, \frac{e^b}{e^a+e^b+e^c}, \frac{e^c}{e^a+e^b+e^c}$. You can first set `vec = vec - max(vec)`, then calculate the same quantity with this new `vec`. Because now at least one component of `vec` is 0, numerical issue is solved.

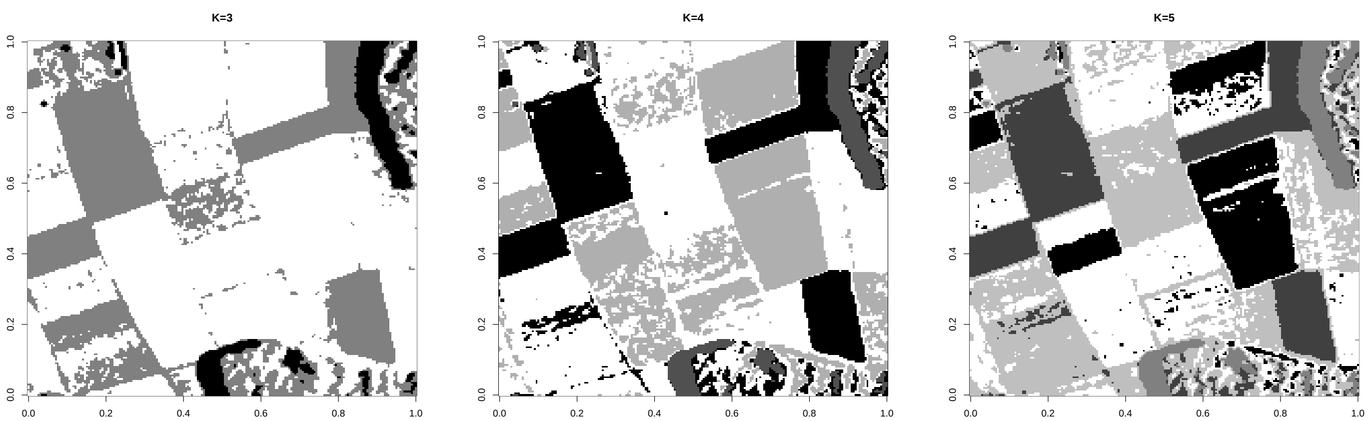

Figure 1: Visualization of the image segmentation, with $K = 3, 4, 5$ and $\tau = 0.01$. The visualization is based on the returned vector `m` from the function `MultinomialEM(H, K, tau)`

□

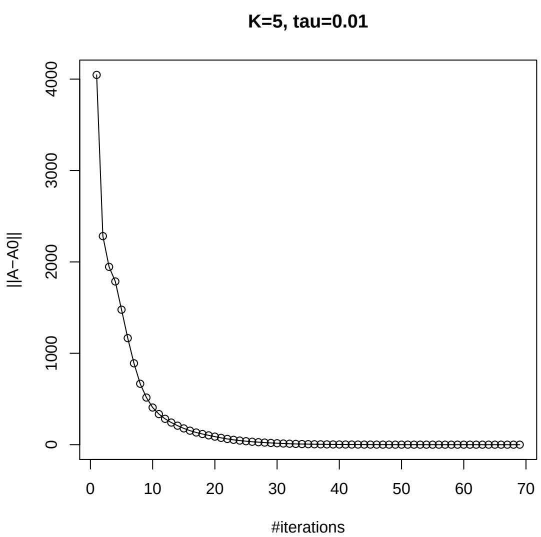

Figure 2: The decreases of $\|A - A_0\|_1$ as number of iterations increases when $K = 5$ and $\tau = 0.01$.