CS498 Homework 6

Xinchen Pan

April 8, 2017

EM Topic models

For this problem, we used NIPS dataset from UCI machine learning repository. We first constructed a 1500×12419 data matrix from the dataset. The row number indicates the number of documents and the column indicates different words. Each cell is the number of count for that word in that document.

We need to cluster the 1500 documents into 30 topics. We used k-means algorithm to get the initials for $\mathbf{p}_j$ and π_j. Both are not given directly, we got from some manipulations from the clustered labels we got from the k-means model result as shown in code.

For E-step, we used

$$p(\delta_{ij} = 1 \,|\, \theta^{(n)}, \mathbf{x}) = \frac{\left[\prod_k p_{j,k}^{x_{i,k}}\right]\pi_j}{\sum_l \left[\prod_k p_{j,k}^{x_{i,k}}\right]\pi_l}$$

to get a 1500×30 probability matrix which we can used for clustering documents.

For M-step, we updated

$$\mathbf{p}_j^{(n+1)} = \frac{\sum_i \mathbf{x}_i w_{ij}}{\sum_i \mathbf{x}_i^T \mathbf{1} w_{ij}}$$

and

$$\pi_j^{(n+1)} = \frac{\sum_i w_{ij}}{N}$$

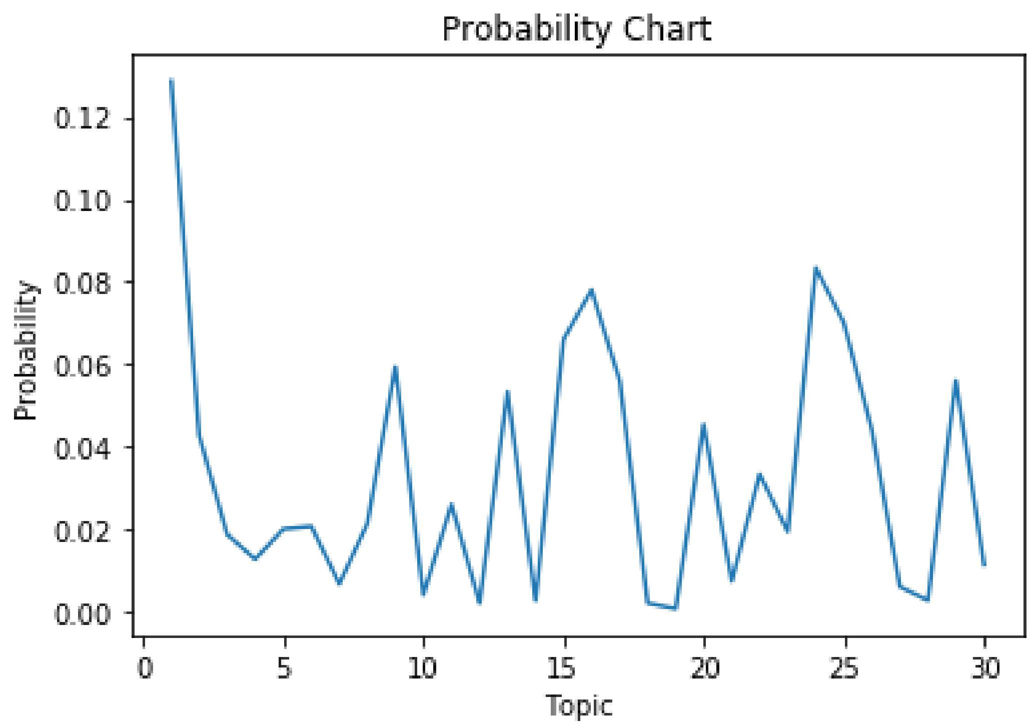
We ran 1000 times this algorithm and checked for Convergence by calculating $Q(\theta, \theta^{(n+1)}) - Q(\theta, \theta^{(n)})$. The expected likelihood with respect to δ is

$$Q(\theta, \theta^{(n)}) = \Big(\sum_{ij} \Big\{\Big[\sum_k x_{i,k} \, \log \, p_{j,k}\Big] + \log \, \pi_j\Big\} w_{ij}\Big)$$

If it is smaller than the threshold we set, the algorithm stopped.

Note in order to deal with the underflow problem, We used expsumlog trick which can be found in the code. We also added some very small number and then scaled back if the word probability is 0.

Below is the graph of the probability with which the topic is selected



Below is the table for top ten frequent words for each topic.

##	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6
## Topic 1	system	model	network	neural	function	input
## Topic 2	unit	network	input	learning	weight	hidden
## Topic 3	learning	action	model	task	control	reinforcement
## Topic 4	algorithm	vector	function	learning	loss	class
## Topic 5	network	unit	input	hidden	output	learning
## Topic 6	weight	network	error	training	set	input
## Topic 7	network	task	neural	learning	training	architecture
## Topic 8	input	network	output	neural	noise	function
## Topic 9	network	training	set	data	neural	error
## Topic 10	classifier	training	network	rbf	set	error
## Topic 11	word	network	recognition	training	system	model
## Topic 12	cell	head	direction	rat	model	angular
## Topic 13	model	data	network	set	neural	parameter
## Topic 14	character	field	system	window	network	input
## Topic 15	data	model	algorithm	set	parameter	point
## Topic 16	network	neural	system	input	function	learning
## Topic 17	learning	algorithm	function	problem	policy	action
## Topic 18	hint	learning	examples	function	error	market
## Topic 19	monte	carlo	player	decision	policy	base
## Topic 20	object	image	network	images	model	recognition
## Topic 21	function	threshold	network	weight	neural	input
## Topic 22	function	set	training	vector	algorithm	error
## Topic 23	speech	network	system	model	input	signal
## Topic 24	function	network	algorithm	learning	neural	model
## Topic 25	cell	model	input	neuron	visual	field
## Topic 26	neuron	network	input	model	neural	synaptic
## Topic 27	david	michael	john	richard	peter	author

```
## Topic 28        eeg component  response       trial  artifact           ica
## Topic 29   learning   network     error      weight  training         input
## Topic 30      model  learning   control    movement     motor       forward
##                Word 7         Word 8      Word 9     Word 10
## Topic 1        signal         output     circuit information
## Topic 2         layer         output     pattern    function
## Topic 3         robot       function      system      states
## Topic 4           set         weight       bound     problem
## Topic 5      function       training     pattern      weight
## Topic 6         noise generalization    function    learning
## Topic 7       control       solution       input     problem
## Topic 8      training            set        data information
## Topic 9         input         output        unit    learning
## Topic 10       neural        problem      center    gaussian
## Topic 11       speech            hmm      neural         set
## Topic 12       system       velocity  mcnaughton      neural
## Topic 13     learning      algorithm    training    function
## Topic 14          net            set        word    training
## Topic 15     learning   distribution      method    function
## Topic 16       weight         output       model        unit
## Topic 17       system        optimal       model      result
## Topic 18  performance         method information     network
## Topic 19         move        rollout     network       trial
## Topic 20         view         system         set     feature
## Topic 21      circuit           size      number      result
## Topic 22       kernel           data     problem  classifier
## Topic 23  recognition         neural      output information
## Topic 24        input        problem         set        data
## Topic 25       cortex    orientation    response     network
## Topic 26       system       function    learning      firing
## Topic 27        index         thomas        eric        paul
## Topic 28         data         single      visual         erp
## Topic 29     function      algorithm      neural         set
## Topic 30        field            arm     dynamic  trajectory
```

Image segmentation using EM

For this problem, we segmented images using a clustering method. We were given 3 images and we need to segment these three images to 10, 20, 50 segments. For the sunset image, we need to segment using five different start points. Thus, for this problem, we had 13 images within the report.

They are

- "A goby on its nest" 10, 20, 50 segments version
- "A strelitzia" 10, 20, 50 segments version
- "Sunset", 10, 20(5 different start points), 50 segments version

I will use "Sunset" image as an example to explain the methods we used. The "Sunset" image is a 330×600 corresponding to height and width image. Each location of the image has a pixel data which contains 3 pixel numbers representing red, green and blue, aka RGB. We formed a dataset of pixel using these data. The dimension of the data is $(330 \times 600 = 198000, 3)$. Then we did normarlization so that each feature of the data has a 0 mean and a 1 standard deviation.

We used cluster centers as initial values for the means and the fraction of points in each cluster as mixture weight initials where we got from k-means algorithm.

For E-step, We firstly calulated

$$p_{(\delta_{ij}=1\,|\theta^{(n)},\mathbf{x})} = \frac{[\exp(-\frac{1}{2}(\mathbf{x}_i - \mu_j)^T(\mathbf{x}_i - \mu_j))]\pi_j}{\sum_k[\exp(-\frac{1}{2}(\mathbf{x}_i - \mu_j)^T(\mathbf{x}_i - \mu_j))]\pi_k} = w_{ij}$$

Then we updated μ amd π at M-step based on

$$\mu_j^{(n+1)} = \frac{\sum_i x_i w_{ij}}{\sum_i w_{ij}}$$

and

$$\pi_j^{(n+1)} = \frac{\sum_i w_{ij}}{N}$$

After updating, we calculated the expected likelihood with respect to δ which is

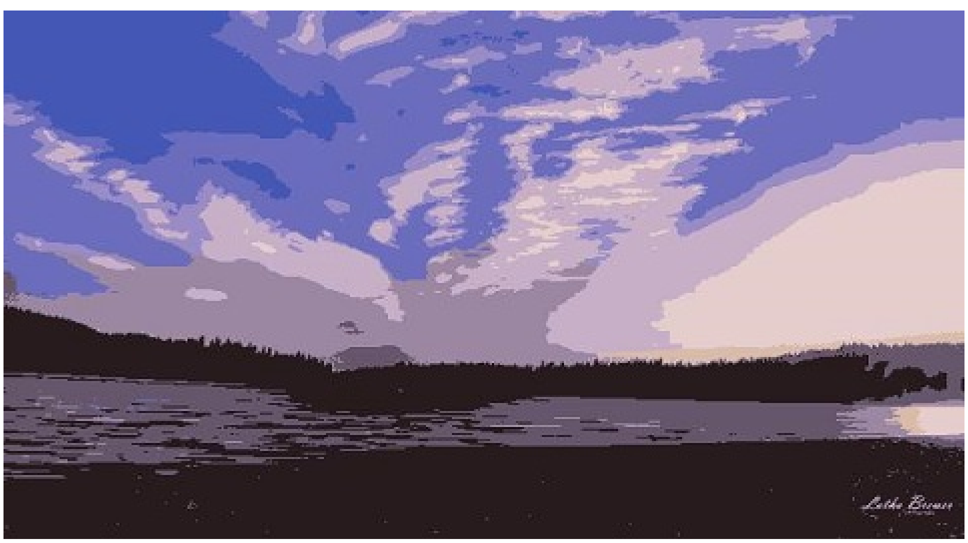
$$Q_{(\theta;\theta^{(n)})} = \Big(\sum_{ij}\big\{\big[\big(-\frac{1}{2}(\mathbf{x}_i - \mu_j)^T(\mathbf{x}_i - \mu_j)\big)\big] + log(\pi_j)\big\}w_{ij} + K\Big)$$

Because of the fairly large computation, We tried to run the algorithm 100 times and set a threshold of 0.0001. If the difference between (Q^n, Q^{n+1}) is smaller than this number, we stopped the alogrithm.

We then replaced the original pixel data by the pixel from μ with the highest value of the posterior probability for that pixel. For example, for all first segment pixel data, we changed the pixel to μ_1, so on and so forth.

Our final result is below

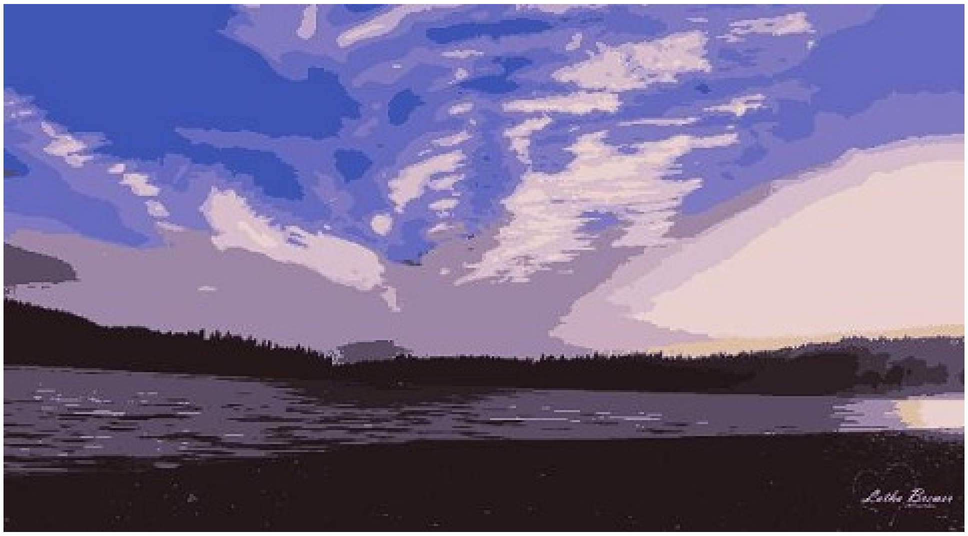
Sunset 10 segments



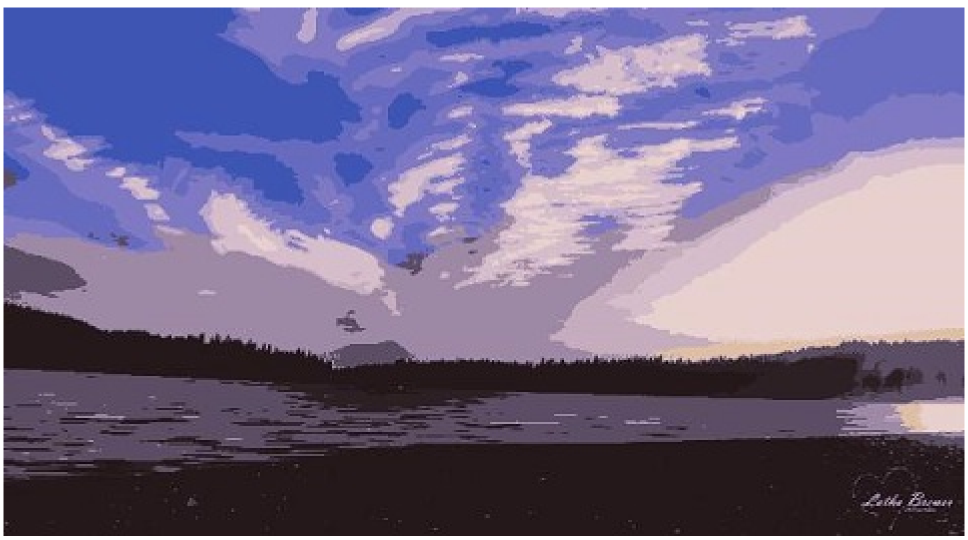
Sunset 10 segments

We used k-means algorithm to produce 5 different start points by setting different seeds. We produced the following five pictures. We can observe that basically these pictures are the same, it is hard to find any differences.

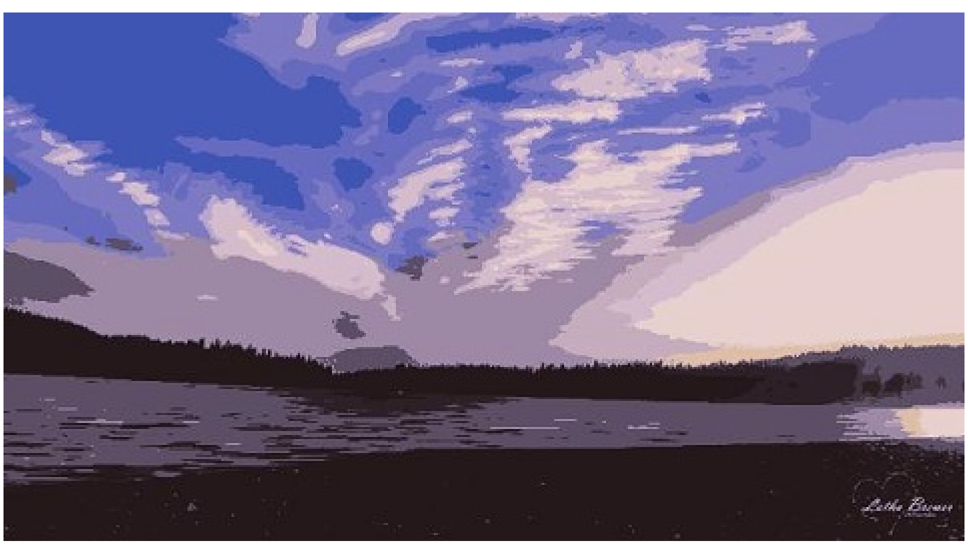
Sunset 20 segments



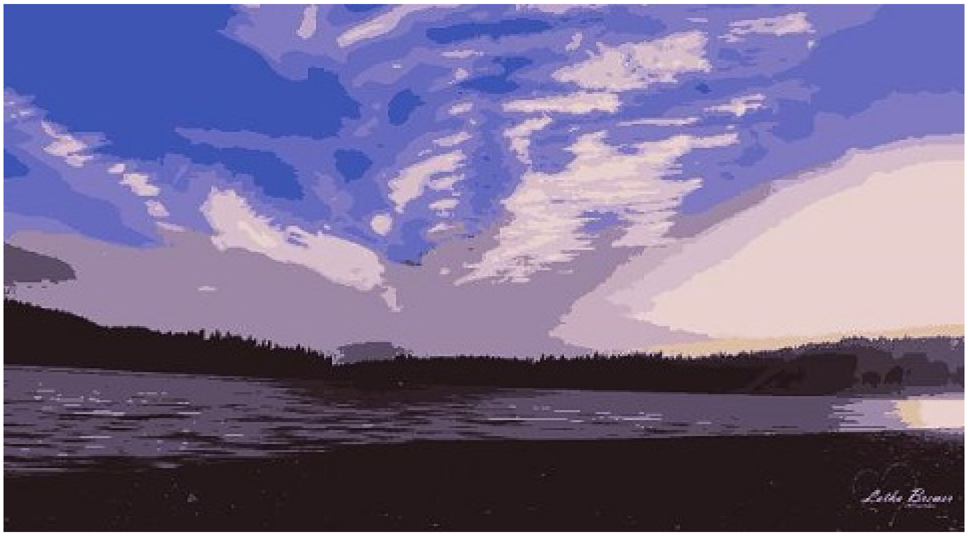
Sunset 20 segments, seed = 25



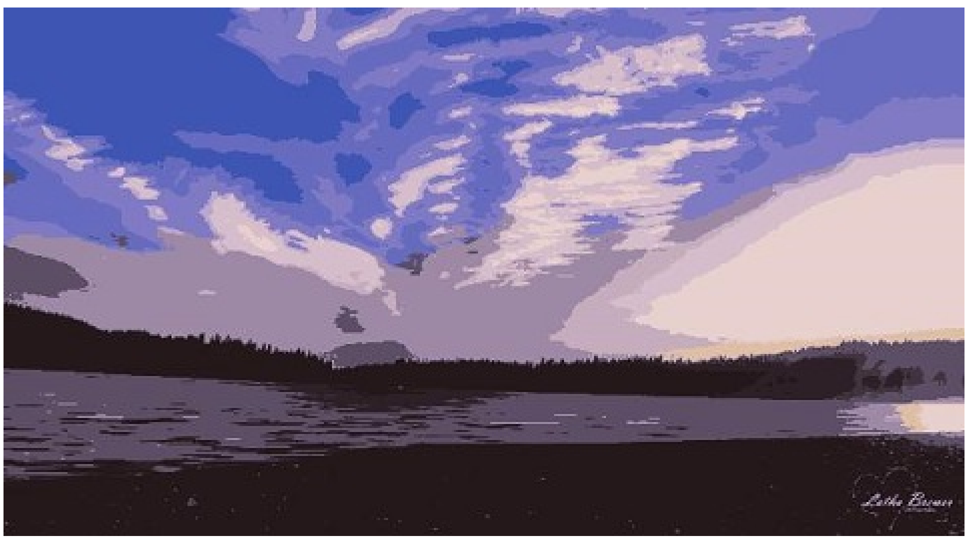
Sunset 20 segments, seed = 33



Sunset 20 segments, seed = 37



Sunset 20 segments, seed = 75

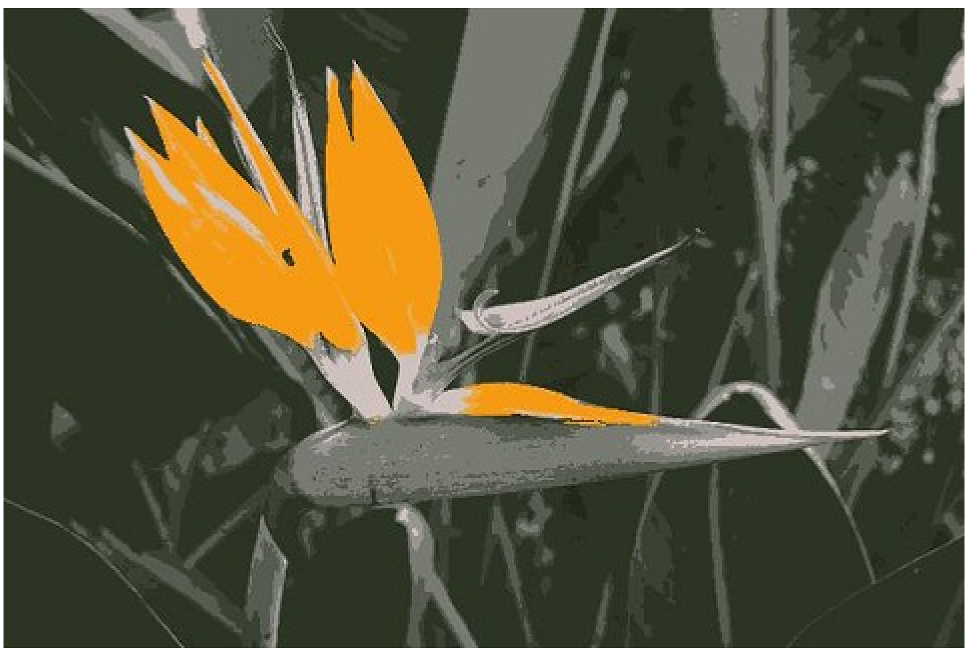


Sunset 20 segments, seed = 83

Sunset 50 Segments

Sunset 50 segments

Flower 10 segments

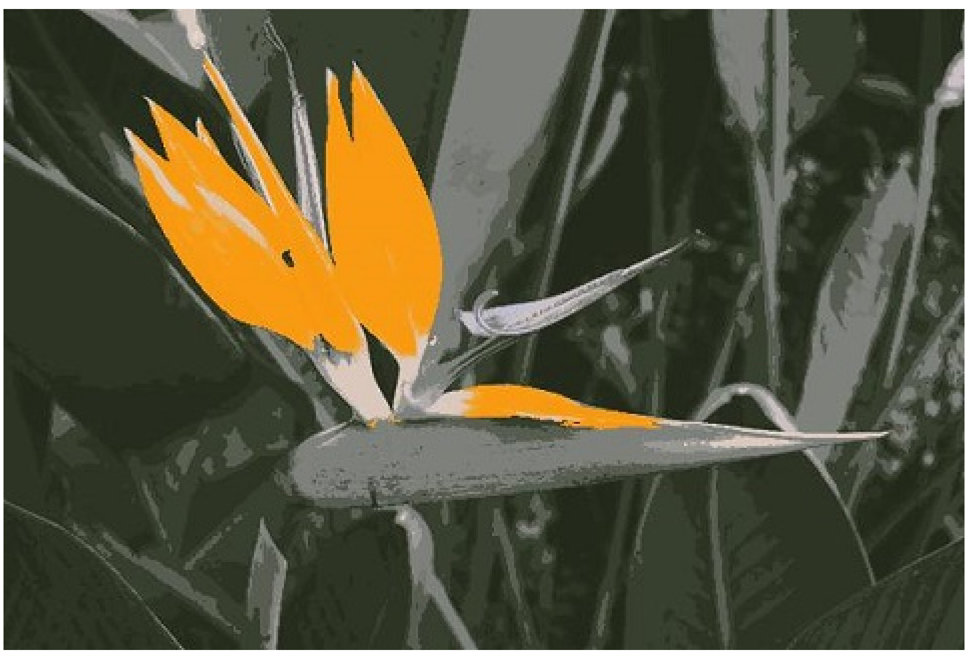


Flower 10 segments

Flower 20 segments

Flower 20 segments

Flower 50 segments



Flower 50 segments

Fish 10 segments

Fish 10 segments

Fish 20 segments

Fish 20 segments

Fish 50 segments

Fish 50 segments