Machine Learning Assignment 2

Yihong Chen April 24, 2018

1 Boosting: From Weak to Strong

Show we can get a strong classifier which achieves zero error on training dataset by boosting some simple thresholding-based decision stumps.

1.1 Weighted error rate of the threshold-based decision stumps

$m_0(s)$ corresponds to the index of the minimum $x^{(i)}$ which is no less than the threshold s i.e.

$$m_0(s) = \quad \arg\min_i x^{(i)} \quad s.t. \quad x^{(i)} \geq s$$

Then the error rate of the classifier $\phi_{s,+}$ is as follows

$$\begin{aligned}
\sum_{i=1}^{m} p_i \mathbf{1}\{\phi_{s,+}(x^{(i)}) \neq y^{(i)}\} &= \sum_{i=1}^{m_0(s)} p_i \mathbf{1}\{\phi_{s,+}(x^{(i)}) \neq y^{(i)}\} + \sum_{i=m_0(s)+1}^{m} p_i \mathbf{1}\{\phi_{s,+}(x^{(i)}) \neq y^{(i)}\} \\
&= \sum_{i=1}^{m_0(s)} p_i \mathbf{1}\{y^{(i)} = -1\} + \sum_{i=m_0(s)+1}^{m} p_i \mathbf{1}\{y^{(i)} = 1\} \\
&= \sum_{i=1}^{m_0(s)} p_i \frac{1 - y^{(i)}}{2} + \sum_{i=m_0(s)+1}^{m} p_i \frac{1 + y^{(i)}}{2} \\
&= \frac{1}{2} - \frac{1}{2}(\sum_{i=1}^{m_0(s)} y^{(i)} p_i - \sum_{i=m_0(s)+1}^{m} y^{(i)} p_i)
\end{aligned} \tag{1}$$

Similarly, we can obtain the error rate of the classifier $\phi_{s,-}$ as follows

$$\sum_{i=1}^{m} p_i \mathbf{1}\{\phi_{s,-}(x^{(i)}) \neq y^{(i)}\} = \frac{1}{2} - \frac{1}{2}(\sum_{i=m_0(s)+1}^{m} y^{(i)} p_i - \sum_{i=1}^{m_0(s)} y^{(i)} p_i) \tag{2}$$

1.2 Margin of the threshold-based decision stumps

Taking the expansion of f, we obtain

$$\begin{aligned}
|f(m_0) - f(m_0+1)| &= |\sum_{i=1}^{m_0} y^{(i)} p_i - \sum_{m_0+1}^{m} y^{(i)} p_i - \sum_{i=1}^{m_0+1} y^{(i)} p_i + \sum_{m_0+2}^{m} y^{(i)} p_i| \\
&= 2|y^{(m_0+1)} p_{m_0+1}| \\
&= 2|p_{m_0+1}|
\end{aligned} \tag{3}$$

Similarly, we have for all $m_0 \in \{0, 1, ..., m\}$

$$\begin{aligned}
|f(m-1) - f(m)| &= 2|p_m| \\
&\cdots \\
|f(0) - f(1)| &= 2|p_1|
\end{aligned} \tag{4}$$

Note that $\sum_{i=1}^{m} p_i = 1$. Hence we sum over all the above equations

$$\sum_{i=1}^{m} |f(i-1) - f(i)| = 2 \tag{5}$$

Using triangle inequality, we obtain

$$\begin{aligned} 2 &= \sum_{i=1}^{m} |f(i-1) - f(i)| \\ &\leq \sum_{i=1}^{m} |f(i-1)| + |f(i)| \\ &\leq \sum_{i=1}^{m} \max_{m_0} |f(m_0)| + \max_{m_0} |f(m_0)| \\ &= 2m \max_{m_0} |f(m_0)| \end{aligned} \tag{6}$$

which leads to

$$\max_{m_0} |f(m_0)| \geq \frac{1}{m} = 2\gamma$$

Hence we can get a weak classifier in each iteration with margin $\gamma = \frac{1}{2m}$

1.3 Number of iterations decision stumps required to achieve zero error

From 1.2 we obtain the threshold-based decision stump can guarantee the margin $\gamma = \frac{1}{2m}$. From theorem 1, we obtain

$$J_t \leq (\sqrt{1 - 4\gamma^2})^t J_0 \tag{7}$$

where J_0 is the initial error rate of the decision stump $J_0 \leq \frac{1}{2} - \gamma$ Hence we have

$$J_t \leq (\sqrt{1 - 4\gamma^2})^t (\frac{1}{2} - \gamma)$$

If we want J_t to be smaller than $\frac{1}{m}$, the upper bound of number of iterations is

$$2 \frac{\log \frac{2}{m-1}}{\log 1 - \frac{1}{m^2}}$$

2 Deep Neural Networks

2.1 configuration for spiral the classification problem

The largest network combined with ReLU and regurlarization gives us a promising result.

activation	ReLU
batchSize	30
learningRate	0.03
regularization	l_2
regularizationRate	0.003
noise	0
networkShape	8,8,8,8,8,8

The classification result is shown as the following figure

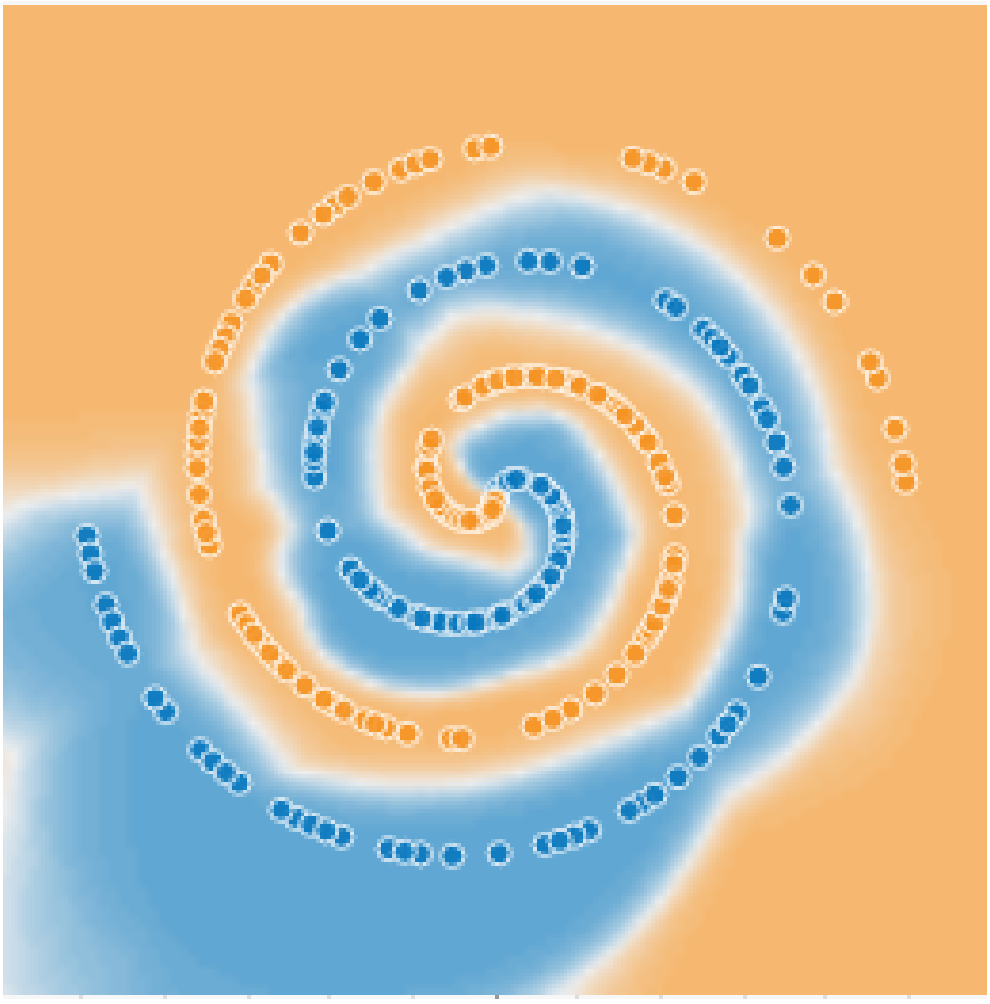

Figure 1: Spiral Classification Result

2.2 How the learning rate, the activation function, the number of hidden layers and the regularization influence the performance and convergence rate

Small learning rate leads to slow convergence rate while it may provide better performance. ReLU leads to faster convergence rate compared to other activation functions. More hidden layers, slower convergence rate. More hidden layers means larger model compacity. If we combine expressive large models with appropriate regularization, we can get better models which generalize well.

2.3 Principles for tuning the network

1. It takes time to tune the network. Please be patient !
2. Understanding overfitting and genralization helps tuning the hypeparams since most of them control model capacity.
3. If the model is complex, regularization can reduce the generalization error.
4. Learning rate is probably the most important hypeparameter.

3 Clustering: Mixture of Multinomials

3.1 MLE for multinomial

With the coefficient unrelated to $\boldsymbol{\mu}$ ignored, the log likelihood is proportional to

$$\log P(\boldsymbol{x}|\boldsymbol{\mu}) \propto \sum_i x_i \log \mu_i \tag{8}$$

Thus, to maximize the log likelihood, we do the following constrained optimization

$$\begin{gathered} \max_{\boldsymbol{\mu}} ll(\boldsymbol{\mu}) = \sum_i x_i \log \mu_i \\ s.t. \sum_i \mu_i = 1 \end{gathered} \tag{9}$$

Using the Lagrange Multiplier, we obtain

$$L(\boldsymbol{\mu}, \lambda) = \sum_i x_i \log \mu_i + \lambda(1 - \sum_i \mu_i) \tag{10}$$

Taking derivatives, we obtain

$$\begin{gathered} \frac{\partial L}{\partial \mu_i} = x_i \frac{1}{\mu_i} - \lambda \\ \frac{\partial L}{\partial \lambda} = 1 - \sum_i \mu_i \end{gathered} \tag{11}$$

By posing all the derivatives to be zero, we get λ and the estimate for $\boldsymbol{\mu}$

$$\begin{gathered} \lambda = \sum_i x_i \\ \mu_i = \frac{x_i}{\sum_i x_i} \end{gathered} \tag{12}$$

3.2 EM for mixture of multinomials

Applying MLE for the parameters $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$, the log-likelihood is as follows

$$\begin{aligned} \log p(D|\boldsymbol{\pi}, \boldsymbol{\mu}) &= \sum_{d=1}^{D} \log(\sum_{k=1}^{K} \pi_k Multi(\boldsymbol{d}|\boldsymbol{\mu_k})) \\ &\propto \sum_{d=1}^{D} \log(\sum_{k=1}^{K} \pi_k \prod_w \mu_{wk}^{T_{dw}}) \end{aligned} \tag{13}$$

Note that we have constraints for both $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$

$$\begin{gathered} \sum_{k=1}^{K} \pi_k = 1 \\ \sum_{w=1}^{W} \mu_{wk} = 1, k \in [1, K] \end{gathered} \tag{14}$$

We optimize the Lagrangian

$$L(\boldsymbol{\pi}, \boldsymbol{\mu}, \lambda, \boldsymbol{\beta}) = \sum_{d=1}^{D} \log(\sum_k \pi_k \prod_w \mu_{wk}^{T_{dw}}) + \lambda(1 - \sum_{k=1}^{K} \pi_k) + \sum_{k=1}^{K} \beta_k (1 - \sum_{w=1}^{W} \mu_{wk})$$

Taking derivatives, we obtain

$$
\begin{aligned}
\frac{\partial L}{\partial \mu_{wk}} &= \sum_{d=1}^{D} \frac{\pi_k \prod_{i \neq w} \mu_{ik}^{T_{di}} T_{dw} \mu_{wk}^{T_{dw}-1}}{\sum_j \pi_j \ \prod_i \mu_{ij}^{T_{di}}} - \beta_k \\
\frac{\partial L}{\partial \pi_k} &= \sum_{d=1}^{D} \frac{\prod_w \mu_{wk}^{T_{dw}}}{\sum_j \pi_j \ \prod_i \mu_{ij}^{T_{di}}} - \lambda \\
\frac{\partial L}{\partial \beta_k} &= 1 - \sum_w \mu_{wk} \\
\frac{\partial L}{\partial \lambda} &= 1 - \sum_k \pi_k
\end{aligned}
\tag{15}
$$

Posing all the derivatives to zero, we obtain

$$
\begin{aligned}
\beta_k &= \sum_w \sum_d \gamma(c_{dk}) T_{dw} \\
\mu_{wk} &= \frac{\sum_d \gamma(c_{dk}) T_{dw}}{\sum_w \sum_d \gamma(c_{dk}) T_{dw}} \\
\lambda &= \sum_k \sum_d \gamma(c_{dk}) \\
\pi_k &= \frac{\sum_d \gamma(c_{dk})}{\sum_k \sum_d \gamma(c_{dk})}
\end{aligned}
\tag{16}
$$

where $\gamma(c_{dk})$ is the responsibilty of the kth topic to the dth document, i.e $P(c_d = k|d)$.

$$
\gamma(c_{dk}) = \frac{\pi_k \prod_w \mu_{wk}^{T_{dw}}}{\sum_j \pi_j \prod_w \mu_{wj}^{T_{dw}}}
$$

The E-step estimate the responsibilities $\boldsymbol{\gamma}$ while the M-step re-estimate the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\pi}$.

3.2.1 Detailed design of the EM algorithm

Key parts of the algorithms are as folows:

Initialization. EM is sensitive to initialization. The $\boldsymbol{\mu}$ and $\boldsymbol{\pi}$ can be initialized as $\frac{T_w}{\sum_w T_w}$ and $\frac{1}{K}$ respectively. This initialization method is quite straightforward but it doesn't work in practice. Random initialization of $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$ may be better.

Log domain calculation. Note that computing the responsibilities involves products of a bunch of small values. We thus perform log domain calculation as follows

$$
\log \hat{\gamma}(c_{dk}) = \log \pi_k + \sum_w T_{dw} \log \mu_{wk} \tag{17}
$$

$$
\log \gamma(c_{dk}) = \log \hat{\gamma}(c_{dk}) - \log(\sum_k \exp \log \hat{\gamma}(c_{dk}))
$$

$$
\log \hat{\mu}_{wk} = \log(\sum_d T_{dw} \exp \log \gamma(c_{dk}))
$$

$$
\log \mu_{wk} = \log \hat{\mu}_{wk} - \log\,(\sum_w \exp \log \hat{\mu}_{wk})
$$

$$
\log \hat{\pi}_k = \log(\sum_d \exp \log \gamma(c_{dk}))
$$

$$\log \pi_k = \log \hat{\pi}_k - \log(\sum_k \exp \log \hat{\pi}_k)$$

Convergence. The EM algorithm monotonically increases the log likelihood of the observed data and converges when the log likelihood of the observed data stays the same.

$$\log p(D|\boldsymbol{\pi}, \boldsymbol{\mu}) \propto \sum_d \log(\sum_k \exp(\log \pi_k + \sum_w T_{dw} \log \mu_{wk}))$$

Prediction. We make predictions using learned $\boldsymbol{\mu}$ and $\boldsymbol{\pi}$. Simply replacing T_{dw} of train data with the one of test data in equation 17 computes the unnormalized log probability of the dth document belonging to the kth topic.

Implementation. The EM algorithm for mixture of multinomials is implemented in python using *numpy*, *scipy* and *sklearn*. Please refer to the source code if you are interested in the implementation.

3.2.2 Experiments

Instead of using the provided matlab format data, we processed the raw data by hand in *Python*. Actually, *sklearn* provides handy functions to load and clean the 20 newsgroups text dataset. The newsgroup-related metadata was stripped. As required by the algorithm, each document was represented in bag-of-words fashion and the whole corpus was converted into the word occurrence matrix. Words with less than 10 occurrences were ignored. The final word occurrence matrix T is 11314×6692. We run EM over the train data.

Results. Larger number of topics chosen, larger the log likelihood. Also large number of topics leads slower training.

K =5 We see that the (talk.religion.misc, alt.atheism, soc.religion.christian) partition corresponds to topic 0; (talk.politics.misc, talk.politics.guns, talk.politics.mideast) partition corresponds to topic 1; the comp and sci correspond to topic 2 and 3. Topic 4 may be others.

0	**christian** believe game use good make year know time **god** say don think people like
1	**gun** say said year **govern state** time think know use **armenian** people right like don
2	look don **card** time problem **program** thank **file** know work like drive window use need
3	**compute** include **data** post like **program** use edu mail avail key inform **encrypt file** com
4	6ei bxn 7ey 2tm 34u g9v giz 1t 75u bhj a86 b8f 1d9 max ax

K =10

0	look make new want use like post don **bike** time think year good **car** know
1	look make use **season** know time don good **player** play like **team** think **game** year
2	say thing want know think key use **problem** people good **work** make time don like
3	think **state** govern work said **armenian** know like space say peopl time use don year
4	check **build** rule info **section** line **gif** format imag **jpeg** use output **program** entrie **file**
5	6ei bxn 7ey 2tm 1t 34u max bhj a86 g9v b8f 75u 1d9 giz ax
6	bit time **control** need thank **disk** problem like work **card** use drive know window scsi
7	include widget use com applic window new **resource** lib **period** list **program** file edu win
8	**gun jesus** law time don know say think like **god** believe use make **christian** people
9	**ftp** like include run mail edu **encrypt** window key **program** inform file use **server** avail

K =20 Some topics are quite similar with slight differences, for example 8, 11 and 19.

0	right **hockey** don people think good like time season new **player** game team play year
1	time run set problem control **disk** use **wire** card like know drive window need work
2	**committe** don **law** report time work jew right like kill year use make **state** people
3	6ei bxn 7ey 1t 2tm giz b8f bhj ax g9v max 75u a86 1d9 34u
4	thing preside want work **jesus** come like time don think said god say know **people**
5	work make problem number rate report case time like space **gun launch** year **car** use
6	question use say think **christian arab** isra god right post know like peopl jew **israel**
7	bit sale offer edu post mail like know price thank new pleas drive use scsi
8	**technology chip** govern don people secure **encrypt jpeg** make like law **image** key use file
9	probe mission univers firearm list window state com orbit edu unit gun file space use
10	mean like time thing atheist think believe don christian religion god say peopl know exist
11	**graphic ftp** inform mail like version includ run avail imag file window use edu program
12	want space need make work know **war** new time like year don people think use
13	car cause know time make day **food** year like good problem don think use peopl
14	**universe christian jesus** people **church** believe word think point time use law space say god
15	good car april men like come edu make year time know peopl **homosexual** use don
16	look right people like time don know **car** use thing say make **bike** good think
17	genocid kill state year **turkey** armenia turk turkish **govern** right greek said armenian time peopl
18	make algorithm clipper people time number **encrypt chip** like bit don use know key secur
19	max use buf int col open printf char check stream line entri **program file** output

K =30

0	someth question like new say don make preside myere people year program know time rule
1	secur key packag data ftp pub program edu technology encrypt inform privaci use file
2	arab law year think know say turkish state kill israel isra said govern use like
3	cub pitch good win make ripem don say key people think time know jesuse
4	window include rocket lunar probe space mission earth new year use orbit work like satellite
5	buy year anyone say thing peopel want new make work like don problem look good
6	help trie look need anyone edu disk program pleas know work drive dos file use
7	video speed chip monitor need thank work port driver problem card scsie like know
8	rule bufe year max return number build open uuencode appear dod check program char entrie
9	process work include send mail software com like line program post time data avail list
10	point need power algorithm make like don des ground bit clipper number know secure time
11	believe said manie good post use people thing god make don know way time say
12	7kn qax b8e air nrhj 6ei 75u 7ey 34u giz 2tm ax bxn 1t
13	tri like event valu problem server com color time applic use file set resourc edu
14	privaci pleas like research list user new file email cancer edu univers anonym inform address
15	qualiti make convert bit display software image version time gif use know don jpeg like
16	mean say good look think need don like henrik phone year armenian work make know
17	women said april trade right like case captain state don
18	need point way high said don want old good use price problem new like
19	philadelphia new calgari boston lose toronto pittsburgh chicago angel power play
20	stl new min win nyr nyi chi det point period team bufe tor think que
21	way look good thing drug car use don bike say want time make
22	war russian look make use way year day work govern said preside say god new
23	source able intern lot world azerbaijan like use power look think year help russian drive
24	people say way hit run player believe time god point church year use good know don like make
25	problem time congress weapon year american right law police like
26	win people time way like play year point want come think say thing game look
27	number make people good edu don nhl league time know like new season play hockey
28	reason mani like thing faith belief believe god exist use people question know atheist jesus
29	image nasa inform server control data good program avail disk ftp support use com version

According to the topic interpretation, it seems that K=5 and K=20 are nice. K=5 means that we have coarse clusters while K=20 means that we have finer clusters. They are both meaningful in terms of the frequent words in them.