Homework 2 of Statistical Machine Learning

Wang Yikai, 2017310740

April 23, 2018

1 Boosting: From Weak to Strong

1. Given that $x^{(1)} > x^{(2)} > \cdots > x^{(m)}$, let $m_0(s) \in 0, 1, ..., m$ equals to an index of x which satisfies $x^{(1)} > x^{(2)} > ... > x^{(m_0(s))} \geq s > x^{(m_0(s)+1)} > \cdots > x^{(m)}$. For the classifier $\phi_{s,-}$, there is:

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

Considering that $\sum_{i=1}^{m} p_i 1\left\{\phi_{s,-}\left(x^{(i)}\right) \neq y^{(i)}\right\} + \sum_{i=1}^{m} p_i 1\left\{\phi_{s,+}\left(x^{(i)}\right) \neq y^{(i)}\right\} = 1$, there is:

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

2. For $f(m_0) = \sum_{i=1}^{m_0} p_i y^{(i)} - \sum_{i=m_0+1}^{m} p_i y^{(i)}$, there is:

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

As p is a distribution of $x^{(1)}, x^{(2)}, \cdots, x^{(m)}$, there exists a p_i satisfying $p_i \geq \frac{1}{m}$. Therefore there exists a m_0 satisfying $|f(m_0) - f(m_0+1)| \geq \frac{2}{m}$. Due to $|f(m_0)| + |f(m_0+1)| \geq |f(m_0) - f(m_0+1)| \geq \frac{2}{m}$, either $|f(m_0)|$ or $|f(m_0+1)|$ is equal or larger than $\frac{1}{m}$.

Therefore,

$$\max_{m_0} |f(m_0)| \geq 2\gamma$$

where $\gamma = \frac{1}{2m}$.

3. Based on the above answer, thresholded decision stumps can guarantee margin γ equal or larger than $\frac{1}{2m}$ on any training set $\{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$.

Due to the Convergence of Boosting, denote J_t as the error rate on training dataset after the T-th iteration, and assume $J_0 = 1$. There is:

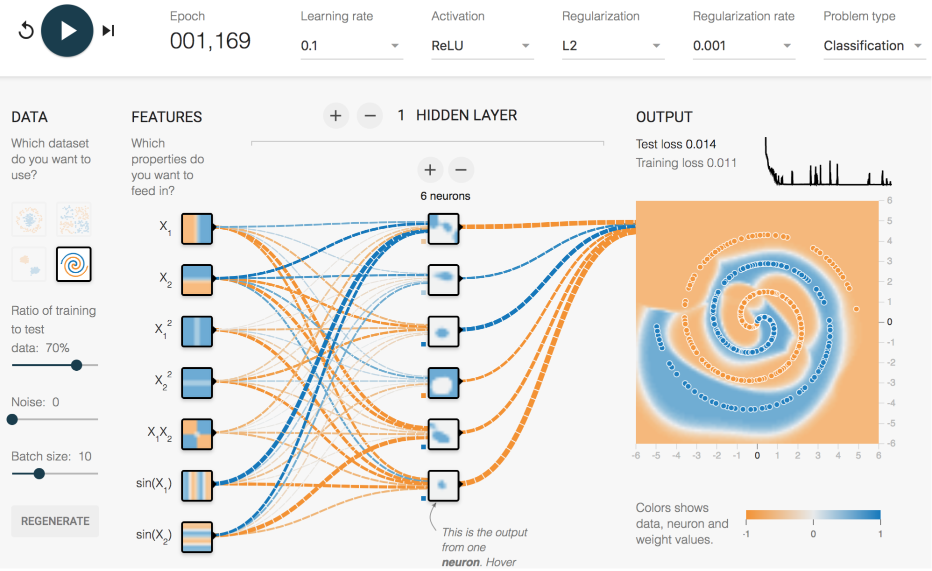
$$J_T \leq \sqrt{1 - 4\gamma_T^2} J_{T-1} \leq \prod_{t=1}^{T} \sqrt{1 - 4\gamma_t^2} \leq \prod_{t=1}^{T} \sqrt{1 - \frac{1}{m^2}} = \left(1 - \frac{1}{m^2}\right)^{\frac{T}{2}}$$

When $J_t < \frac{1}{m}$, we have zero training error. Let $(1 - \frac{1}{m^2})^{\frac{T}{2}} < \frac{1}{m}$, there is:

$$T > \frac{-2\log(m)}{\log\left(1 - \frac{1}{m^2}\right)}$$

2 Deep Neural Networks

1. As shown in figure 1, when more various features but fewer layers are put into the network, the network can achieve a good nonlinear performance.



Epoch 001,169

Learning rate 0.1

Activation ReLU

Regularization L2

Regularization rate 0.001

Problem type Classification

Ratio of training to test data: 70%

Noise: 0

Batch size: 10

1 HIDDEN LAYER

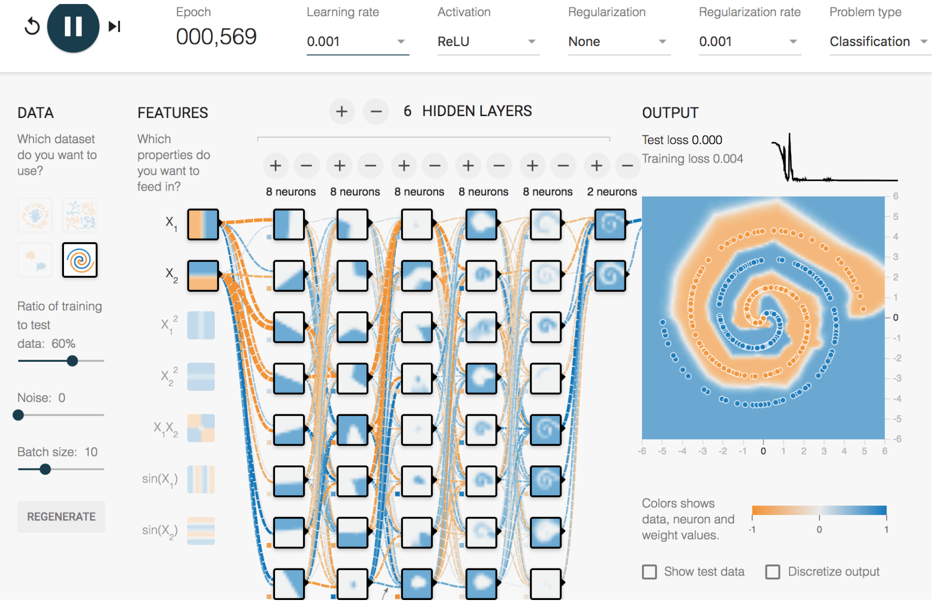
6 neurons

Test loss 0.014

Training loss 0.011

DATA
Which dataset do you want to use?
REGENERATE
FEATURES
Which properties do you want to feed in?
X_1
X_2
X_1^2
X_2^2
X_1X_2
$\sin(X_1)$
$\sin(X_2)$
+
−
This is the output from one neuron. Hover
OUTPUT
6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6
-6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6
Colors shows data, neuron and weight values.
-1 0 1

Figure 1: Use more features but fewer layers



Epoch 000,569

Learning rate 0.001

Activation ReLU

Regularization None

Regularization rate 0.001

Problem type Classification

DATA

Which dataset do you want to use?

Ratio of training to test data: 60%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

6 HIDDEN LAYERS

OUTPUT

Test loss 0.000

Training loss 0.004

Colors shows data, neuron and weight values.

□ Show test data □ Discretize output

Figure 2: Use fewer features but more layers

As shown in figure 2, when only the two basic features but use more layers are used, the network can also achieve a good nonlinear performance.

During the training, the learning rate is 0.1 for the first 400 epochs, 0.03 for the next 100 epochs, and 0.01 for the rest epochs. The batch size is set as 10 and the ratio of training to test is 60%. The activation function is ReLU, and there is no regularization.

2. In this dataset, regularization term is not a must. As shown in figure 2, test loss reaches 0 meaning that there is no overfitting, though there is no regularization.

Also, in this dataset ReLU function can achieve a better performance compared with other activation functions such as Tanh, Sigmoid and Linear(I use all activation functions training the network for 500 epochs, the testing loss with ReLU function is the lowest).

At the beginning of the training, the learning rate is set as 0.1 in order to speed up the learning process. At the end of the training, the learning rate needs diminishing so that the loss will not oscillate. Also, during this training, setting the batch size to 10 seems a fairly good choice since it can avoid large variance, avoid local minimum, and the training time is also acceptable.

3. During the training, there are several details need to be adjusted carefully:

(1) Usually ReLU is the best choice for activation function.

(2) Choosing a good learning rate is important, sometimes we can use an auto decaying learning rate, or just diminish the learning rate after some epochs.

(3) Batch normalization can accelerate training process.

(4) The initialization needs also adjusting, sometimes use uniform initialization and sometimes Xavier Initialization or He Initialization works better.

(5) We can use regularization(such as L1, L2) or Dropout(default 0.5, sometimes 0.2 if the model is not very complicated) to avoid overfitting.

(6) Choose a good optimizer method from various methods such as Adagrad, RMSProp.

3 Clustering: Mixture of Multinomials

3.1 MLE for multinomial

For multinomial distribution:

$$P(\boldsymbol{x}|\boldsymbol{\mu}) = \frac{n!}{\prod_{i=1}^{d} x_i!} \prod_{i=1}^{d} \mu_i^{x_i}, \; i = 1, \cdots, d$$

where $x_i \in \mathbb{N}$, $\sum_{i=1}^{d} x_i = n$, and the parameter $\boldsymbol{\mu} = (\mu_i)_{i=1}^{d}$, $0 < \mu_i < 1$, $\sum_{i=1}^{d} \mu_i = 1$.

The log likelihood for $P(\boldsymbol{x}|\boldsymbol{\mu})$:

$$\log P(\boldsymbol{x}|\boldsymbol{\mu}) = \log(n!) - \sum_{i=1}^{d} \log(x_i!) + \sum_{i=1}^{d} x_i \log(\mu_i)$$

Maximize the log likelihood w.r.t. $\boldsymbol{\mu}$, and consider the constraint $\sum_{i=1}^{d} \mu_i = 1$. Therefore, first derive the Lagrange function of $\log P(\boldsymbol{x}|\boldsymbol{\mu})$:

$$L(\boldsymbol{\mu}, \beta) = \log(n!) - \sum_{i=1}^{d} \log(x_i!) + \sum_{i=1}^{d} x_i \log(\mu_i) - \beta \left(\sum_{i=1}^{d} \mu_i - 1 \right)$$

where β is Lagrange multipliers.

Solve the gradient of $L(\boldsymbol{\mu}, \beta)$ w.r.t. $\boldsymbol{\mu}$, $\boldsymbol{\alpha}$ and the derivative w.r.t. β are:

$$\nabla_{\boldsymbol{\mu}} L(\boldsymbol{\mu}, \beta) = \left[\frac{x_1}{\mu_1} - \beta, \frac{x_2}{\mu_2} - \beta, \cdots, \frac{x_d}{\mu_d} - \beta \right]^T$$

$$\frac{\partial L(\boldsymbol{\mu}, \beta)}{\beta} = -\sum_{i=1}^{d} \mu_i + 1$$

Let them equal to zero, there are $\mu_i = \frac{x_i}{\beta}$, $i = 1, 2, \cdots, d$, and $\sum_{i=1}^{d} \mu_i = 1$. Therefore:

$$\sum_{i=1}^{d} \mu_i = \sum_{i=1}^{d} \frac{x_i}{\beta} = 1 \Rightarrow \sum_{i=1}^{d} x_i = \beta = n$$

So the maximum-likelihood estimator for the parameter $\boldsymbol{\mu}$ is:

$$\mu_i = \frac{x_i}{n}, \; i = 1, 2, \cdots, d$$

which also satisfies $0 < \mu_i < 1$.

For N samples, the gradient of $\tilde{L}(\boldsymbol{\mu}, \beta)$ w.r.t. $\boldsymbol{\mu}$, $\boldsymbol{\alpha}$ and the derivative w.r.t. β are:

$$\begin{aligned}
\nabla_{\boldsymbol{\mu}} \tilde{L}(\boldsymbol{\mu}, \beta) &= \left[\sum_{n=1}^{N} \frac{x_{n1}}{\mu_1} - \beta, \sum_{n=1}^{N} \frac{x_{n2}}{\mu_2} - \beta, \cdots, \sum_{n=1}^{N} \frac{x_{nd}}{\mu_d} - \beta \right]^T \\
\frac{\partial \tilde{L}(\boldsymbol{\mu}, \beta)}{\beta} &= -\sum_{i=1}^{d} \mu_i + 1
\end{aligned}$$

Therefore for N samples, the maximum-likelihood estimator for $\boldsymbol{\mu}$ is:

$$\mu_i = \frac{\sum_{n=1}^{N} x_{ni}}{\sum_{j=1}^{d} \sum_{n=1}^{N} x_{nj}} = \frac{1}{Nn} \sum_{n=1}^{N} x_{ni}, \; i = 1, 2, \cdots, d$$

which satisfies $0 < \mu_i < 1$.

3.2 EM for mixture of multinomials

The log likelihood of D documents is:

$$\begin{aligned}
J(\boldsymbol{\pi}, \boldsymbol{\mu}) &= \log \prod_{d=1}^{D} P(d) \\
&= \sum_{d=1}^{D} \log \sum_{k=1}^{K} P(d|c_d = k) P(c_d = k) \\
&= \sum_{d=1}^{D} \log \frac{n_d!}{\prod_{\omega=1}^{W} T_{d\omega}!} \sum_{k=1}^{K} \pi_k \prod_{\omega=1}^{W} \mu_{\omega k}^{T_{d\omega}} \\
&= \sum_{d=1}^{D} \log \frac{n_d!}{\prod_{\omega=1}^{W} T_{d\omega}!} + \sum_{d=1}^{D} \log \sum_{k=1}^{K} \pi_k \prod_{\omega=1}^{W} \mu_{\omega k}^{T_{d\omega}} \\
&= \sum_{d=1}^{D} \log \frac{n_d!}{\prod_{\omega=1}^{W} T_{d\omega}!} + \sum_{d=1}^{D} \log \sum_{k=1}^{K} q(z_d = k) \frac{\pi_k \prod_{\omega=1}^{W} \mu_{\omega k}^{T_{d\omega}}}{q(z_d = k)} \\
&\geq \sum_{d=1}^{D} \log \frac{n_d!}{\prod_{\omega=1}^{W} T_{d\omega}!} + \sum_{d=1}^{D} \sum_{k=1}^{K} q(z_d = k) \log \frac{\pi_k \prod_{\omega=1}^{W} \mu_{\omega k}^{T_{d\omega}}}{q(z_d = k)}
\end{aligned}$$

where $q(z_d = k) = P(c_d = k|d)$ is the posterior distribution, and the inequality holds due to the Jensen inequality.

Denote $\tilde{J}(q, \boldsymbol{\pi}, \boldsymbol{\mu})$ as:

$$\tilde{J}(q, \boldsymbol{\pi}, \boldsymbol{\mu}) = \sum_{d=1}^{D} \sum_{k=1}^{K} q(z_d = k) \log \frac{\pi_k \prod_{\omega=1}^{W} \mu_{\omega k}^{T_{d\omega}}}{q(z_d = k)}$$

The **E-step** is to update the posterior of the $(t+1)$th iteration $q^{(t+1)}(z_d = k)$, $d = 1, \cdots, D$, $k = 1, \cdots, K$.

Denote $q^{(t+1)}(z_d = k)$ as $\gamma_d^{(t+1)}(k)$, where $\sum_{k=1}^K \gamma_d^{(t+1)}(k) = 1$. There is:

$$\gamma_d^{(t+1)}(k) = P(c_d = k|d) = \frac{P(c_d = k, d)}{\sum_{k=1}^K P(c_d = k, d)} = \frac{\pi_k^{(t)} \prod_{\omega=1}^W \mu_{\omega k}^{(t)T_{d\omega}}}{\sum_{k=1}^K \pi_k^{(t)} \prod_{\omega=1}^W \mu_{\omega k}^{(t)T_{d\omega}}}$$

where $k = 1, \cdots, K$.

$\gamma_d^{(t+1)}(k)$ can be written as below in order to avoid overflow of $\mu_{\omega k}^{(t)T_{d\omega}}$:

$$\begin{aligned}
\gamma_d^{(t+1)}(k) &= \exp\left[\log \gamma_d^{(t+1)}(k)\right] \\
&= \exp\left[\log\left(\pi_k^{(t)} \prod_{\omega=1}^W \mu_{\omega k}^{(t)T_{d\omega}}\right) - \log \sum_{k=1}^K \left(\pi_k^{(t)} \prod_{\omega=1}^W \mu_{\omega k}^{(t)T_{d\omega}}\right)\right] \\
&= \exp\left[a_k - \log \sum_{k=1}^K \exp\left(a_k\right)\right] \\
&= \exp\left[a_k - \log \sum_{k=1}^K \exp\left(a_k - max_a\right) \exp\left(max_a\right)\right] \\
&= \exp\left[a_k - max_a - \log \sum_{k=1}^K \exp\left(a_k - max_a\right)\right]
\end{aligned}$$

where $a_k = \log\left(\pi_k^{(t)} \prod_{\omega=1}^W \mu_{\omega k}^{(t)T_{d\omega}}\right) = \log \pi_k^{(t)} + \sum_{\omega=1}^W T_{d\omega} \log \mu_{\omega k}^{(t)}$, $max_a = \max_k a$.

The **M-step** is to update the parameters of the $(t+1)$th iteration $\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}$:

$$\begin{aligned}
\boldsymbol{\pi}^{(t+1)} &= \arg\max_{\boldsymbol{\pi}} \tilde{J}\left(q^{(t+1)}, \boldsymbol{\pi}, \boldsymbol{\mu}^{(t)}\right) \\
\boldsymbol{\mu}^{(t+1)} &= \arg\max_{\boldsymbol{\mu}} \tilde{J}\left(q^{(t+1)}, \boldsymbol{\pi}^{(t+1)}, \boldsymbol{\mu}\right)
\end{aligned}$$

Considering the constraints $\sum_{k=1}^K \pi_k = 1$, $\sum_{\omega=1}^W \mu_{\omega k} = 1$, $k = 1, \cdots, K$, use Lagrange functions $L(\boldsymbol{\pi}, \boldsymbol{\mu})$ to solve the maximum problems. There are:

$$\begin{aligned}
\sum_{d=1}^D \frac{\gamma_d^{(t+1)}(k)}{\pi_k} - \beta = 0, \sum_{k=1}^K \pi_k = 1 \;&\Rightarrow\; \pi_k^{(t+1)} = \frac{\sum_{d=1}^D \gamma_d^{(t+1)}(k)}{\sum_{k=1}^K \sum_{d=1}^D \gamma_d^{(t+1)}(k)} \\
\sum_{d=1}^D \frac{\gamma_d^{(t+1)}(k) T_{d\omega}}{\mu_{\omega k}} - \beta = 0, \sum_{\omega=1}^W \mu_{\omega k} = 1 \;&\Rightarrow\; \mu_{\omega k}^{(t+1)} = \frac{\sum_{d=1}^D \gamma_d^{(t+1)}(k) T_{d\omega}}{\sum_{\omega=1}^W \sum_{d=1}^D \gamma_d^{(t+1)}(k) T_{d\omega}}
\end{aligned}$$

where $k = 1, \cdots, K$, $\omega = 1, \cdots, W$.

My program depends on Python packages **numpy**, **os**, **tqdm** and **sklearn**. There are two Python files: **main.py** contains reading data, removing stopping words, EM procedure and results; **evaluate.py** is to evaluate the results by using Adjusted Mutual Information.

The results are shown in **outputs/**, containing topics and 5 most frequency words for $K = 5, 10, 20, 30$ respectively and the evaluation result. The evaluation result shows that the best

K value for this dataset is 20, because the Adjusted Mutual Information score when $K = 20$ is the highest among the scores when $K = 5, 10, 20, 30$ (AMI = 0.038, 0.050, 0.056, 0.055 respectively).

Most frequent words:

K=5:

For topic 0:

’writes’ ’people’ ’article’ ’don’ ’time’

For topic 1:

’people’ ’god’ ’don’ ’writes’ ’article’

For topic 2:

’writes’ ’article’ ’don’ ’people’ ’good’

For topic 3:

’writes’ ’article’ ’don’ ’good’ ’people’

For topic 4:

’don’ ’writes’ ’key’ ’people’ ’article’

K=10:

For topic 0:

’writes’ ’article’ ’people’ ’time’ ’don’

For topic 1:

’writes’ ’don’ ’article’ ’people’ ’windows’

For topic 2:

’writes’ ’article’ ’people’ ’god’ ’don’

For topic 3:

’writes’ ’article’ ’scsi’ ’don’ ’people’

For topic 4:

’god’ ’writes’ ’people’ ’don’ ’article’

For topic 5:

’db’ ’people’ ’article’ ’writes’ ’space’

For topic 6:

’don’ ’people’ ’writes’ ’article’ ’file’

For topic 7:

’writes’ ’article’ ’don’ ’apr’ ’ca’

For topic 8:

’writes’ ’article’ ’people’ ’don’ ’space’

For topic 9:

’writes’ ’people’ ’don’ ’article’ ’good’

K=20:

For topic 0:

’writes’ ’article’ ’don’ ’good’ ’apr’

For topic 1:

’writes’ ’article’ ’people’ ’don’ ’gun’

For topic 2:

’people’ ’window’ ’don’ ’writes’ ’article’

For topic 3:

’writes’ ’article’ ’people’ ’time’ ’good’

For topic 4:

’writes’ ’article’ ’don’ ’people’ ’good’

For topic 5:

’writes’ ’article’ ’space’ ’don’ ’data’

For topic 6:

’writes’ ’article’ ’file’ ’output’ ’don’

For topic 7:

’scsi’ ’drive’ ’mb’ ’card’ ’bit’

For topic 8:

’god’ ’mr’ ’don’ ’stephanopoulos’ ’writes’

For topic 9:

’db’ ’writes’ ’article’ ’people’ ’time’

For topic 10:

’writes’ ’article’ ’don’ ’time’ ’people’

For topic 11:

’people’ ’writes’ ’article’ ’hockey’ ’don’

For topic 12:

’writes’ ’key’ ’encryption’ ’article’ ’chip’

For topic 13:

’people’ ’space’ ’don’ ’writes’ ’time’

For topic 14:

’writes’ ’people’ ’article’ ’don’ ’time’

For topic 15:

’don’ ’people’ ’writes’ ’jpeg’ ’good’

For topic 16:

’god’ ’people’ ’writes’ ’article’ ’don’

For topic 17:

’people’ ’writes’ ’god’ ’article’ ’file’

For topic 18:

’writes’ ’jews’ ’article’ ’turkish’ ’years’

For topic 19:

’writes’ ’article’ ’don’ ’good’ ’people’

K=30:

For topic 0:

’writes’ ’don’ ’game’ ’year’ ’people’

For topic 1:

’writes’ ’information’ ’don’ ’time’ ’people’

For topic 2:

’god’ ’people’ ’writes’ ’article’ ’don’

For topic 3:

’writes’ ’article’ ’don’ ’people’ ’ca’

For topic 4:

’writes’ ’article’ ’time’ ’don’ ’people’

For topic 5:

’db’ ’mov’ ’space’ ’bh’ ’cs’

For topic 6:

’writes’ ’article’ ’don’ ’good’ ’time’

For topic 7:

’people’ ’god’ ’writes’ ’article’ ’don’

For topic 8:

’wire’ ’ground’ ’wiring’ ’people’ ’good’

For topic 9:

’writes’ ’article’ ’ftp’ ’people’ ’don’

For topic 10:

’writes’ ’ms’ ’myers’ ’article’ ’don’

For topic 11:

’mr’ ’stephanopoulos’ ’don’ ’people’ ’writes’

For topic 12:

’writes’ ’people’ ’article’ ’don’ ’time’

For topic 13:

’file’ ’output’ ’don’ ’people’ ’program’

For topic 14:

’writes’ ’don’ ’article’ ’time’ ’system’

For topic 15:

’writes’ ’article’ ’people’ ’apr’ ’don’

For topic 16:

’writes’ ’article’ ’turkish’ ’people’ ’don’

For topic 17:

’don’ ’writes’ ’article’ ’israel’ ’people’

For topic 18:

’writes’ ’article’ ’don’ ’time’ ’good’

For topic 19:

’people’ ’jesus’ ’writes’ ’key’ ’time’

For topic 20:

’drive’ ’writes’ ’don’ ’mb’ ’scsi’

For topic 21:

’scsi’ ’article’ ’writes’ ’dos’ ’gm’

For topic 22:

’writes’ ’article’ ’don’ ’people’ ’apr’

For topic 23:

’writes’ ’article’ ’god’ ’file’ ’drive’

For topic 24:

’writes’ ’drive’ ’article’ ’windows’ ’scsi’

For topic 25:

’god’ ’writes’ ’article’ ’people’ ’don’

For topic 26:

’people’ ’don’ ’writes’ ’image’ ’time’

For topic 27:

’writes’ ’card’ ’article’ ’drive’ ’time’

For topic 28:

’writes’ ’people’ ’don’ ’article’ ’apr’

For topic 29:

’encryption’ ’writes’ ’don’ ’key’ ’article’