CS 289A Homework 3

Jiaying Shi

February 28, 2015

1 Problem 1

(a) The sampled mean of sampled data is denoted as $\bar{X}_1$and $\bar{X}_2$.Running the program, we will get:

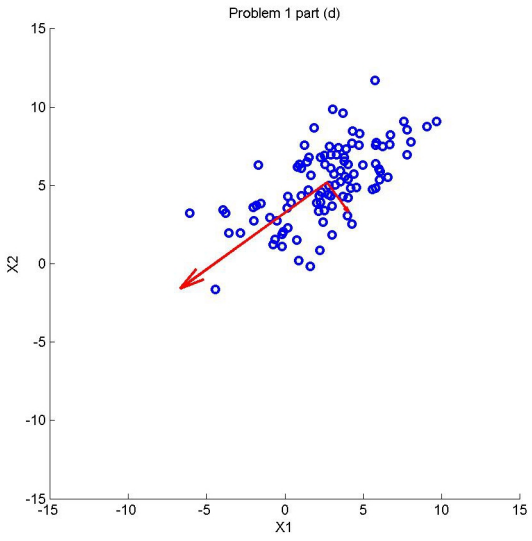
$$\bar{X_1} = 2.6676,\ \bar{X_2} = 5.2043$$

(b) The covariance matrix of the sampled data is:

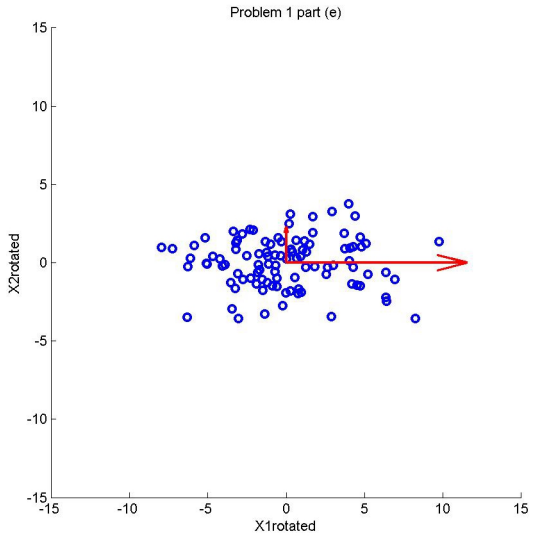
$$\hat{\Sigma} = \begin{bmatrix} 9.2961 & 4.8775 \\ 4.8775 & 6.1028 \end{bmatrix}$$

(c) The first eigenvalue is $\lambda_1 = 2.5672$ and the corresponding eigenvector is $v_1 = \begin{bmatrix} 0.5869 & -0.8097 \end{bmatrix}^T$.
The second eigenvalue is $\lambda_1 = 12.8316$ and the corresponding eigenvector is $v_1 = \begin{bmatrix} -0.8097 & -5869 \end{bmatrix}^T$.

(d) The graph of all the 100 data points and the eigen-vectors are shown in the following graph:



(e) The rotated data points and eigenvectors are shown in the following graph:



2 Problem 2

(a) If the covariance matrix of a multivariate Gaussian distribution is singular, there exist a Y such that Y can be expressed as:

$$Y = \sum_{i=1}^{n} a_i X_i,\ a_i \in \mathrm{R},\ \text{and not all } a_i = 0$$

where:

$$\mathrm{E}(Y) = a_0 \text{ and } \mathrm{Var}(Y) = 0$$

That means if the coordinates of random vector X is linear dependent, the covariance matrix will be singular.
When the covariance matrix, with $rank(\Sigma) = k < n$, is singular , there are two ways to convert X to X'without loss of information where $\Sigma_{X'}^{-1}$ exists. The first way is to choose k linear independent coordinates from X to compose X'. The second way is to add $X_s$to X and let $X' = X + \epsilon X_s$ where $X_s \sim N(0, I)$and $X_s$is independent of X. In this case $\Sigma_{X'} = \Sigma + \epsilon I$ is non-singular. Since we know the distribution of X_s, we can always recover X.

(b) To prove the conclusion, the first thing to notice is that Σ. Then by spectral theorem, Σ can be decomposed as $\Sigma = U\Lambda U^T$. When Σ is non-singular, we have $\Sigma^{-1} = U\Lambda^{-1}U^T$, where

$$\Lambda = diag\left(\begin{bmatrix} \lambda_1 & \lambda_2 & \dots & \lambda_n \end{bmatrix}^T\right),\ \lambda_i > 0 \text{ for all } i = 1, \dots n$$

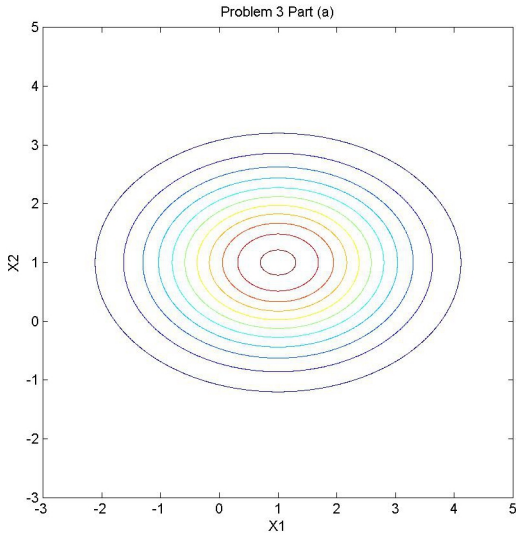
Let $\Gamma = diag\left(\begin{bmatrix} \lambda_1^{-\frac{1}{2}} & \lambda_2^{-\frac{1}{2}} & \dots & \lambda_n^{-\frac{1}{2}} \end{bmatrix}^T\right)$. Then we have, $\Sigma^{-1} = (U\Gamma)(U\Gamma)^T = AA^T$. Thus, we have $x^T\Sigma^{-1}x = x^TAA^Tx = \langle Ax, Ax\rangle = \frac{2}{2}$, where $A = U\Gamma$.

(c) When X is a zero mean multivariate normal random variable, Ax is a linear transformation of any vector $x \in R^n$. The linear transformation rotate vector x and scale x in a way such that AX is a standard multivariate normal random variable. And $\|Ax\|_2^2$ is the square of the norm of the vector after the linear transformation.

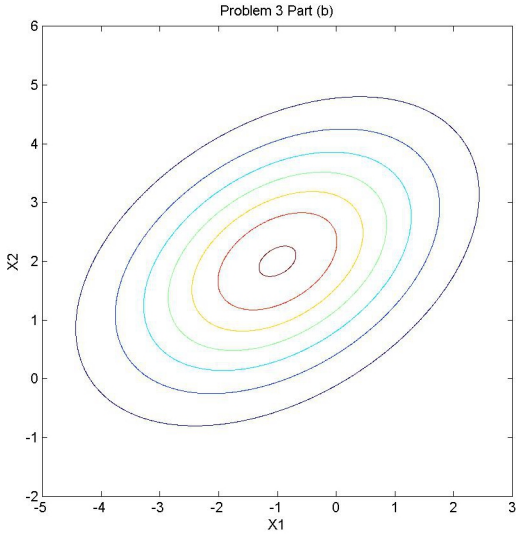
(d) Since $\|Ax\|_2^2 = x^T\Sigma^{-1}x$, when $\|x\| = 1$, the maximum value for $\|Ax\|_2^2$ is $\frac{1}{\lambda_{min}}$ where λ_{min} is the smallest of all the eigenvalues. The minimum value for $\|Ax\|_2^2$ is $\frac{1}{\lambda_{max}}$ where λ_{max} is the largest of all the eigenvalues. When the coordinates of X are mutually independent, the maximum value of $\|Ax\|_2^2$ is the inverse of the largest variance and the minimum value of $\|Ax\|_2^2$ is the inverse of the smallest variance. It is the square of the length semimajor axis and the square of the length of semiminor axis of an elipsoid transformed from a ball with radius 1 in R^n .

3 Problem 3

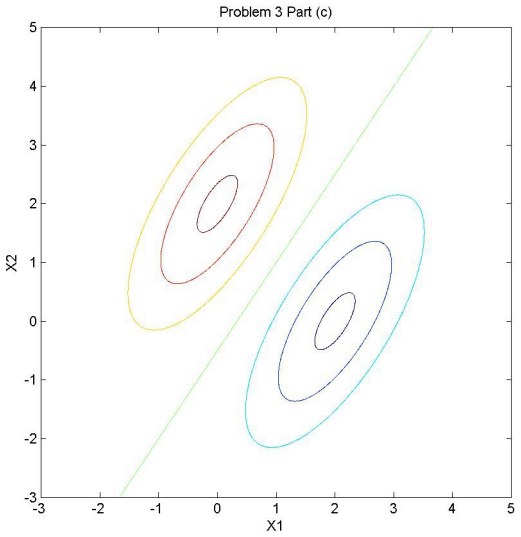
(a)



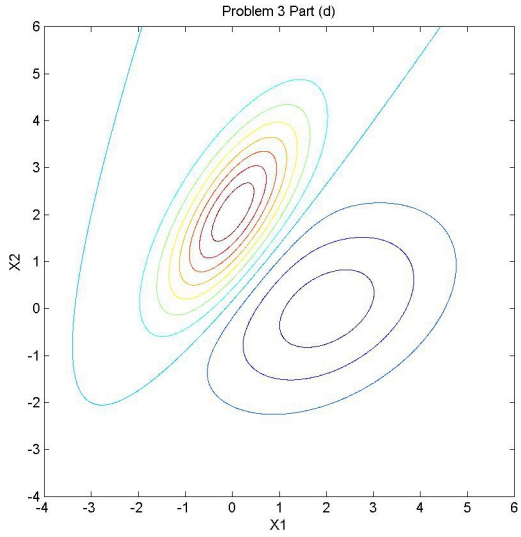
(b)



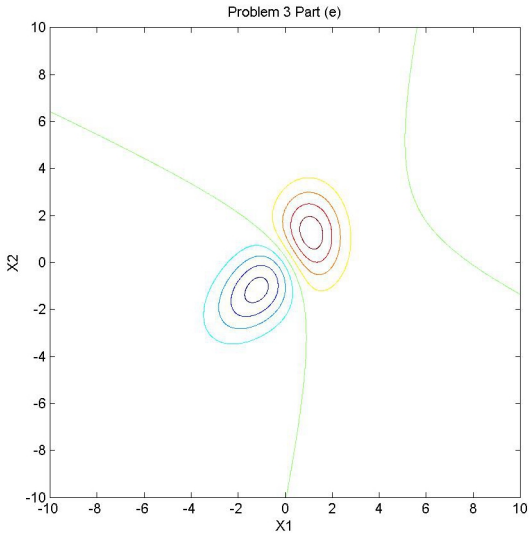
(c)



(d)



(e)



4 Problem 4

(a) The maximal likelihood estimator of the mean is :

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

Since $E(\hat{\mu}) = \frac{1}{n}\sum_{i=1}^{n} E(X_i) = E(X_i) = \mu$ the estimator is unbiased.
The maximum likelihood estimator of the covariance matrix is:

$$\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n} (X_i - \hat{\mu})(X_i - \hat{\mu})^T$$

The expected value of $\hat{\Sigma}$is

$$E\left(\hat{\Sigma}\right) = E\left(\frac{1}{n}\sum_{i=1}^{n} (X_i - \hat{\mu})(X_i - \hat{\mu})^T\right) = E\left(\frac{1}{n}\sum_{i=1}^{n} (X_i - \mu + \mu - \hat{\mu})(X_i - \mu + \mu - \hat{\mu})^T\right)$$

$$= E\left(\frac{1}{n}\sum_{i=1}^{n} (X_i - \mu)(X_i - \mu)^T - (\hat{\mu} - \mu)(\hat{\mu} - \mu)^T\right) = \Sigma - E\left((\hat{\mu} - \mu)(\hat{\mu} - \mu)^T\right) \neq \Sigma$$

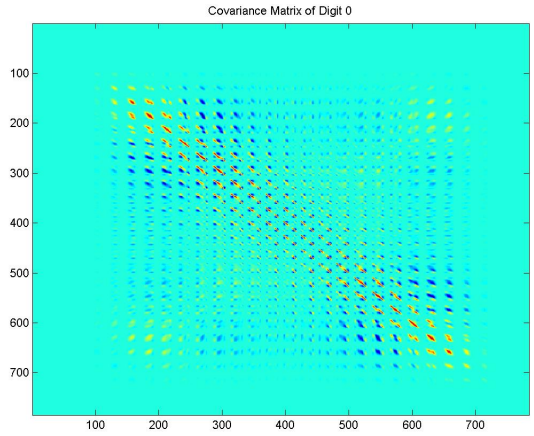
Thus, the maximum likelihood estimator for variance is biased.

(b) The prior distribution can be obtained by $Pr(D_i) = \frac{\#digit\ i-1}{\#of\ training\ set}$. For the training set, the probability for the ten classes are:

Digit	**0**	**1**	**2**	**3**	**4**
$Pr(D_i)$	**0.09795**	**0.1125**	**0.1000**	**0.1016**	**0.09793**
Digit	**5**	**6**	**7**	**8**	**9**
$Pr(D_i)$	**0.09023**	**0.09908**	**0.1039**	**0.09755**	**0.09918**

We can see tha the distribution is nearly uniform.

(c) The covariance matrix for digit 0 is shown in the following figure.



From the above figure, we can see that the covariance matrix is singular. For some points, it is always white so the value for some pixel are determinant instead of stochastic. Moreover, from the figure, we can see that the diagonal component are larger than the off-diagonal components, which is in accord with $Var(X) \geq Cov(X, Y)$.

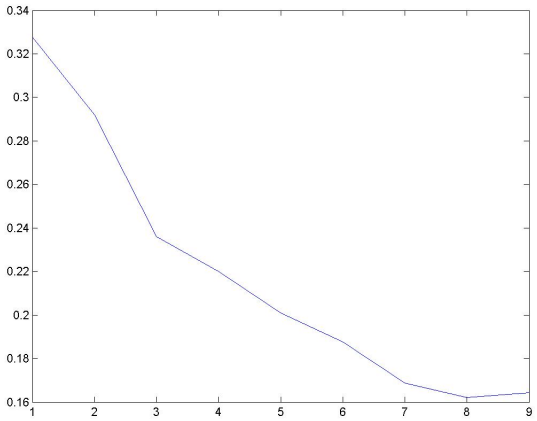
(d)

i) When implementing the algorithm, for digit i, since the covariance matrix $\Sigma_{overall}$ is singular, we need to add a perturbation matrix ϵI to the covariance to make the inverse of it exist. Parameter ϵ is obtained through cross validation. Since we are required to use the raw data, there are 784 features in the problem. Thus for any digit i, the Gaussian distribution is:

$$f_{X|D_i}(x) = (2\pi)^{-392} |\Sigma_{overall} + \epsilon I|^{-\frac{1}{2}} exp\left(-\frac{1}{2}(x-\mu_i)^T (\Sigma_{overall} + \epsilon I)^{-1}(x-\mu_i)\right)$$

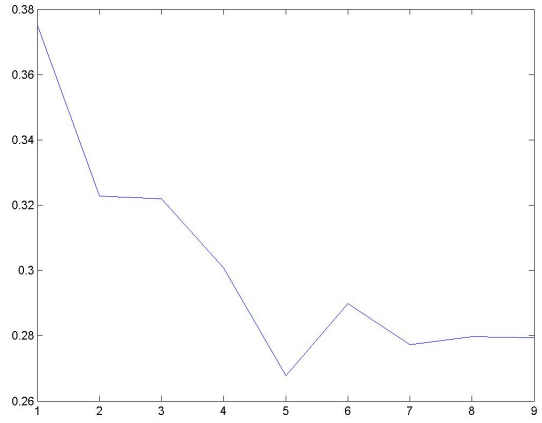
where $(2\pi)^{-392}$is a very small number. Thus $f_{X|D_i}(x)$ is a very small number. When using the build-in function in Matlab to calculate probability, for each point, a inverse of a large matrix and determinant of the matrix is required to be calculated, which makes the program slow. To avoid these problems, by that for different classes the covariance matrix is the same and the $log(x)$is a monotone function, we can compare $\frac{1}{2}(x-\mu_i)^T (\Sigma_{overall} + \epsilon I)^{-1}(x-\mu_i) - log(Pr(Di))$ for different classes instead of the actual probabilities. Since the covariance matrices for different classes are the same, the axises of ellipsoids for the level sets of different classes are parallel with each other. By symmetry, the decision boundary between different classes are hyperplanes.

The error rate for different sample size is shown in the following graph.



ii) When the covariance matrices for different classes are different, the decision boundaries are no longer hyperplanes. Because the symmetry in i) no longer exists. The

The error rate for different sample size is shown in the following graph.



There are jumps in the error rates when the sample size increases. The reason for that is the samples are randomly drawed from the training set and the points in each class are not uniformly distributed.

iii) The results for part a is better than those of part b. I think there are for two reasons that. First, when using different covariance matrices, the decision boundrary is nonlinear, there may be over fitting problems. Moreover, the covariance matrix for each class is singular, when adding a perturbation matrix, there may be large numerical errors when the conditional numbers of the covariance matrices are large.

iv) The accuracy rate for the Kaggle test is 84.88%. I did not use additional featurizer.

e) The features that I added are listed here:

> http,www, link, discount, cost, specials, earn, free, luxury, guaranteed, website, edu, investment, snoring, loss, investment, valium, viagra, vicodin, Dear, price, url, re :, meeting, dollar, fyi, hello, deal, sales, cash, bonus, cheap, rate, click, insurance, preview, order, membership, resume, conference, thanks, investment, $, cc: .

The accuracy rate for Kaggle test is 80.123%

5 Problem 5

Since $\bar{\mathbf{x}} = \frac{1}{n}\sum_i \mathbf{x_i} = 0$ we have:

$$\mathbf{X^T 1 = 0}$$

Given $\lambda > 0$, the loss function has positive definite Hessian matrix, thus it is convex. To find the optimizer, we can take partial derivatives:

$$\frac{\partial J(\mathbf{w}, w_0)}{\partial \mathbf{w}} = 2\left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)\mathbf{w} + 2\mathbf{X}^T\mathbf{y} - 2w_0\mathbf{X}^T\mathbf{1} = 2\left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)\mathbf{w} + 2\mathbf{X}^T\mathbf{y} = \mathbf{0}$$

$$\frac{\partial J(\mathbf{w}, w_0)}{\partial w_0} = 2nw_0 - 2\mathbf{w}^T\mathbf{X}^T\mathbf{1} - \mathbf{y}^T\mathbf{1} = 2nw_0 - \sum_i y_i = 0$$

Solving the above equations, we will get the optimizer:

$$\hat{w}_0 = \bar{y}$$

$$\hat{\mathbf{w}} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y}$$

6 Problem 6

The log likelihood is

$$l(\omega_0, \omega_1) = log\left(\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(y_i - \omega_0 - \omega_1 x_i)^2}{2\sigma^2}\right)\right) = nlog\frac{1}{\sqrt{2\pi}} - nlog\sigma - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \omega_0 - \omega_1 x_i)^2$$

The log likelihood function is concave in ω_0 and ω_1. To obtain ω_0 and ω_1, we can take the derivative:

$$\frac{\partial l(\omega_0, \omega_1)}{\partial \omega_0} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i - \omega_0 - \omega_1 x_i) = 0$$

$$\frac{\partial l(\omega_0, \omega_1)}{\partial \omega_1} = \frac{1}{\sigma^2}\sum_{i=1}^{n} x_i(y_i - \omega_0 - \omega_1 x_i) = 0$$

Then we get:

$$\omega_0 = \bar{y} - \omega_1\bar{x} \approx E(Y) - \omega_1 E(X)$$

$$\omega_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\sum_i x_i y_i - n\bar{x}\bar{y}}{\sum_i x_i^2 - n\bar{x}^2} \approx \frac{cov(X, Y)}{Var(X)}$$

7 Problem 7

(a) We know that $Pr(Y = 1|X = 0) = Pr(Y = -1|X = 0) = \frac{1}{2}$. Then $Pr(Y = 0|X = 0) = 1 - Pr(Y = 1|X = 0) - Pr(Y = -1|X = 0) = 0$. Thus we know $Pr(X = 0, Y = 0) = 0$.

Since $Pr(X = 0) = \frac{1}{2}$ and $Pr(Y = 0) = \frac{1}{2}$ we know $Pr(X = 0, Y = 0) \neq Pr(X = 0)Pr(Y = 0)$. Hence, X and Y are not independent.
Moreover, by the definition of conditional probability, $Pr(X = 0, Y = 1) = Pr(X = 0, Y = -1) = Pr(Y = 1|X = 0)Pr(X = 0) = \frac{1}{4}$. Similarly we have

$$Pr(X = 1, Y = 0) = Pr(X = -1, Y = 0) = Pr(X = 1|Y = 0)Pr(Y = 0) = \frac{1}{4}.$$

Since $\sum_i \sum_j Pr(X = i, Y = j) = 1$, we have the joint distribution :

$$Pr(X = i, Y = j) = \begin{cases} 0 & if\ ij \neq 0\ or\ i = j = 0 \\ \frac{1}{4} & otherwise \end{cases} \quad i, j\ takes\ value\ from\ \{-1, 0, 1\}$$

Then we have:

$$E(XY) = 0$$

Calculating the marginal distribution, we will have:

$$Pr(X = 1) = Pr(X = -1) = \frac{1}{4}$$

$$Pr(Y=1) = Pr(Y=-1) = \frac{1}{4}$$

Then we have:

$$E(X) = 0$$

$$E(Y) = 0$$

Thus by the definition of covariance, we have:

$$cov(X,Y) = E(XY) - E(X)E(Y) = 0$$

which means the two random variables are uncorrelated. X and Y are uncorrelated but not independent.

(b) By symmetry, to see whether X , Y and Z are pairwise independent, we only need to check whether X and Y are independent.

$$Pr(X=1) = Pr(B_1=1, B_2=0) + Pr(B_1=0, B_2=1) = \frac{1}{2}$$

Similarly, we have $Pr(X=i) = Pr(Y=i) = Pr(Z=i) = \frac{1}{2}$ where $i = 0, 1$. The joint distribution of X and Y is:

$$Pr(X=1, Y=1) = Pr(B_1=1, B_2=0, B_3=1) + Pr(B_1=0, B_2=1, B_3=0) = \frac{1}{4}$$

$$Pr(X=1, Y=0) = Pr(B_1=1, B_2=0, B_3=0) + Pr(B_1=0, B_2=1, B_3=1) = \frac{1}{4}$$

$$Pr(X=0, Y=1) = Pr(B_1=1, B_2=1, B_3=0) + Pr(B_1=0, B_2=0, B_3=1) = \frac{1}{4}$$

$$Pr(X=0, Y=0) = Pr(B_1=1, B_2=1, B_3=1) + Pr(B_1=0, B_2=0, B_3=0) = \frac{1}{4}$$

Thus when i and j take value from $\{0, 1\}$, we have:

$$Pr(X=i, Y=j) = Pr(X=i)\,Pr(Y=j)$$

So X , Y and Z are pairwise independent.
To see X , Y and Z are not mutually independent, we only need to see that

$$Pr(X=1, Y=1, Z=1) = 0 \neq Pr(X=1)\,Pr(Y=1)\,Pr(Z=1)$$

8 Problem 8

(a) I think one problem of Daniel's method is that he used the milliseconds since the midnight. In that case, even the spam is received with in a few minutes, the value for the new feature is march larger than the other features. In SVM, the norm of the θ will be minimized. The corresponding value of θ_n of the new feature is very small. Then comparing with other features, the new feature will make little contribution in the classification. I think one way to deal with this issue is to scale the value of the new feature to make it smaller. Another method is to set a threshold of the time that the email is received. For example, if the email is received within 5 minutes from the midnight, then the new feature is 1 otherwise it is 0.

(b) Content-based spam filters have efficiency and scalability issues when there are tons of emails to be classified. Moreover, some words with high probability appearing in spams may be hams for a certainty type of users. So general content based spam filter may not work well for certainty types of users. I think one way to solve this problem is to update the features dynamically. Other anti-spam techniques may include domain names filtering enforcing RFC standards, rule based filtering and so on.