CS57800 Statistical Machine Learning
Homework 3

Ting Zhang
School of Industiral Engineering
zhan1013@purdue.edu

November 11, 2015

1 Foundations

1. The VC dimension of C is $2n + 1$. For a convex polygon with n vertices, the VC dimension of it is $2n + 1$. Then, it is clear that for any polygon with n vertices, the VC dimension is at least $2n + 1$. Also, it can be shown that for $2n + 2$ points, there is always one point that can not be labeled correctly.

2. To show it is a convex function, we can compute the second derivative of the function.

$$\frac{dl}{d\hat{y}} = \frac{1}{\log 2}[\frac{-ye^{-y\hat{y}}}{(1+e^{-y\hat{y}})\log 2}] = \frac{1}{(\log 2)^2}(-1 + \frac{y}{1+e^{-y\hat{y}}})$$
$$\frac{d^2l}{d\hat{y}^2} = \frac{1}{(\log 2)^2}[0 + y * (-2) * (\frac{1}{1+e^{-y\hat{y}}})^2 * (-y) * e^{-y\hat{y}}] = \frac{2y^2e^{-y\hat{y}}}{(\log 2)^2(1+e^{-y\hat{y}})} > 0$$

Since the second derivative of this function is greater than 0, then this function is a convex function.

3. The training error of the final hypothesis can be formulated as

$$TrainingError(H_{final}) \leq \prod_t [2\sqrt{\epsilon_t(1-\epsilon_t)}] \leq e^{-2\sum_t(\gamma)^2}.$$

Since we want final hypothesis has training error of 0, then $e^{-2\sum_t(\gamma)^2} < 1/n$. This equation can be reformulated as $-2T(1-\gamma)^2 < -\ln n$. Therefore, we have $T > \frac{\ln n}{2(1-\gamma)^2}$.

4. The decision boundary of each weak hypothesis is a axis-parallel line, as shown in Figure 1. Positive points are represented with "+", and negative points are represented with "*". Then the process of doing Adaboost is as following:
 (1) First iteration $t = 1$, find the decision boundary that makes the least number of mistakes. In first iteration, each example is uniformlly distributed, with $D_1(i) = 0.1$. Here it found $x_1 > 6$ and $x_2 > 6$, with 3 mistakes, including example 1, 5 and 10. Therefore, the error is $\epsilon_1 = 0.3$. The weak hypothesis is $h_1(x_i) : sign(x_1 - 6) * sign(x_2 - 6)$.

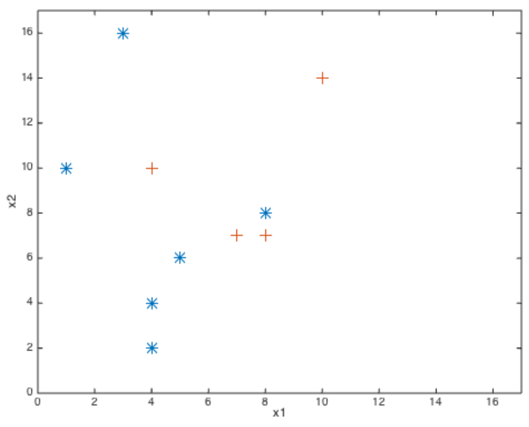


Figure 1: Examples shown in 2d representation.

(2) Second iteration $t = 2$, first, update the distribution of each example. To update the distribution, we need to follow these euqations:

$$\alpha_1 = 0.5 * \ln(\frac{1-\epsilon_1}{\epsilon_1}) = 0.5 * \ln(\frac{1-0.3}{0.3}) = 0.424 e_{-\alpha_1} = 0.654 e_{\alpha_1} = 1.53$$

So, the distribution of the correctly classified examples $i = 2, 3, 4, 6, 7, 8, 9$ becomes

$$D_2(i) = \frac{D_1(i) * e_{-\alpha_1}}{Z_1} = \frac{0.0654}{Z_1}$$

and the distribution of the wrongly classified examples $i = 1, 5, 10$ becomes

$$D_2(i) = \frac{D_1(i) * e_{\alpha_1}}{Z_1} = \frac{0.153}{Z_1}$$

Since $\sum_{i=1}^{10} D_2(i) = 1$, we have $(7 * 0.0654 + 3 * 0.153)/Z_1 = 1$. So, $Z_1 = 0.9162$. Therefore, the distribution of the examples now become

$$D_2(i) = \begin{cases} 0.0714 & \text{if } i = 2, 3, 4, 6, 7, 8, 9 \\ 0.1668 & \text{if } i = 1, 5, 10 \end{cases}$$

Then, finding the weak hypothesis that has the smallest error $\sum_{i=1} D_2(i)$, where i is the index of wrongly classified example. Then, it find the weak hypothesis to be $h_2(x_i) : sign(x_1 - 16) * sign(x_2 - 16)$, with four mistakes, including example 3, 6, 7, 9. And the error is $\epsilon_2 = 0.2856$. We can also compute $\alpha_2 = 0.5 * \ln(\frac{1-\epsilon_2}{\epsilon_2}) = 0.458$.
Then, the final hypothesis is

$$\begin{aligned} H_{final}(x) &= sign(\sum_t \alpha_t h_t(x)) \\ &= sign(0.0.424 * sign(x_1 - 6) * sign(x_2 - 6) + 0.458 * sign(x_1 - 16) * sign(x_2 - 16)) \end{aligned}$$

5. To verify a kernel function, we need to proof the positive semi-definite property of it.

$$\begin{aligned}
&\iint f(\vec{x})K(\vec{x},\vec{y})f(\vec{y})d\vec{x}d\vec{y} \\
&= \iint f(\vec{x})[\alpha K_1(\vec{x},\vec{y}) + \beta K_2(\vec{x},\vec{y})]f(\vec{y})d\vec{x}d\vec{y} \\
&= \iint \alpha f(\vec{x})K_1(\vec{x},\vec{y})f(\vec{y})d\vec{x}d\vec{y} + \beta f(\vec{x})K_2(\vec{x},\vec{y})f(\vec{y})d\vec{x}d\vec{y} \\
&> 0*0 + 0*0 \\
&= 0
\end{aligned}$$

   Because, $K_1(\vec{x},\vec{y})$ and $K_2(\vec{x},\vec{y})$ are positive semi-definite, α and β are all positive.

6. ξ is defined as the margin of the soft SVM. When an example is classified correctly, $0 \leq \xi \leq 1$; while if an example is classified wrongly, then $\xi > 1$. Therefore, with M examples, there are at most M mistakes. Since $\xi > 1$, then $\sum_{i=1}^{M}\xi_i > M*1 = M$. Therefore, $\sum_{i=1}^{M}\xi_i$ is the upper bound on the training error of the classifier.

2 Programming Report

In this assignment, Gradient decent algorithm with hinge lost is implemented. Two forms of regularization term are applied, including the "l1" and "l2" norm of weights. As the last homework, three features are experimented in this task, including the unigram, bigram and both unigram and bigram feature set. Four performance of these features and different regularization terms are measured, including the accuracy, precision, recall and f-score. Detail explanation is presented below.

2.1 Gradient Decent

In this gradient decent algorithm, hinge lost is applied with two forms of regularization, "l1" and "l2" norm. Hinge lost can be computed using the following equation:

$$loss(y, f(x)) = max(0, 1 - yf(x))$$

where $f(x) = wx + b$. And the general form of norm for the weight, w, can be represented as:

$$||w||_p = (\sum_d |w_d|^p)^{\frac{1}{p}}$$

Then, for "l1" norm, it would be $||w|| = (\sum_d |w_d|)$. And for "l2" norm, it would be $||w||_2 = \sqrt{\sum_d |w_d|^2}$.
In the gradient decent algorithm, we are going to minimize the objective function, $loss(y, f(x)) + \lambda R(w)$, where $R(w)$ is the regularization term. To compute the gradient of this objective function, we can get:

* loss function

$$g(w) = \begin{cases} 0 & \text{if } y(wx+b) > 1 \\ yx & \text{otherwise} \end{cases}$$

* l1: $g(w) = sign(w)$

* l2: $g(w) = w$

Therefore, the algorithm can be written as in Algorithm 1.

Algorithm 1 Gradient Decent algorithm with hinge lost

```
1:  procedure GRADIENT DECENT
2:      $w = (0, 0, ..., 0), b = 0$
3:      for iter = 1:MaxIterations do
4:          $g = (0, 0, ..., 0), gb = 0$
5:          for $(x, y) \in D$ do
6:              if $y(wx + b) \leq 1$ then
7:                  $g = g + yx$
8:                  $gb = gb + y$
9:              end if
10:         end for
11:         if regularization == "l1" then
12:             $g = g - \lambda sign(w)$
13:         else if regularization == "l2" then
14:             $g = g - \lambda w$
15:         end if
16:         $w = w + \eta g$
17:         $b = b + \eta gb$
18:     end for
19: end procedure
```

2.2 Experiment

Experiments are conducted by selecting the correct parameters and with different feature set and regularization terms.

2.2.1 Parameters

For the gradient decent algorithm, there are several parameters, including the stepSize, regularization coefficient, λ, and the number of iterations. In this task, the number of iterations is used as a hyper-parameter that tuned with a validation set. The parameter stepSize is selected empirically, the value of objective function is checked when selecting this parameter. When the objective function start to increase with a relatively small number of iterations, the stepSize is considered as too large. Multiple values are experimented and 0.0001 is selected. The regularization coefficient is used to control the complexity of the trained model. It is selected as 1.

Table 1: Test performance of I1 norm for different feature set

	Accuracy	Precision	Recall	F-score
Unigram	0.75	0.76	0.72	0.74
Bigram	0.61	0.82	0.28	0.42
Both	0.75	0.78	0.67	0.72

Table 2: Test performance of I2 norm for different feature set

	Accuracy	Precision	Recall	F-score
Unigram	0.74	0.73	0.75	0.74
Bigram	0.67	0.65	0.76	0.70
Both	0.75	0.73	0.78	0.75

2.2.2 Results

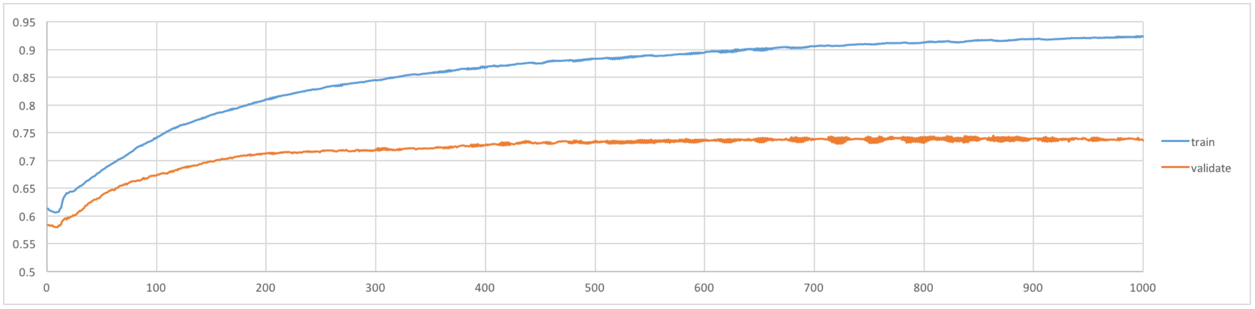
Experiments are conducted with 1000 iterations as maximum. The preformance are shown below.

* "l1": Figure 2 shows the accuracy of training and validation for each feature set with "l1" regularization, over 1000 iterations. The precision, recall and f-score for training and validation set for each feature set is shown in Figure 3, 4, and 5, respectively. From Figure 2, the number of iterations for the test set is decided. For unigram feature set, after 600 iterations, the accuracy for validation set is not growing. To avoid overfitting, 600 iterations is selected for the test test. For bigram feature set, the validation accuracy starts decreasing around 400 iterations, therefore, 400 iterations is selected for bigram feature set. For the one using both feature set, the validation accuracy decreases around 300 iterations. Therefore, 300 is selected for it. The test performance for unigram, bigram and both feature set is shown in Table 1.

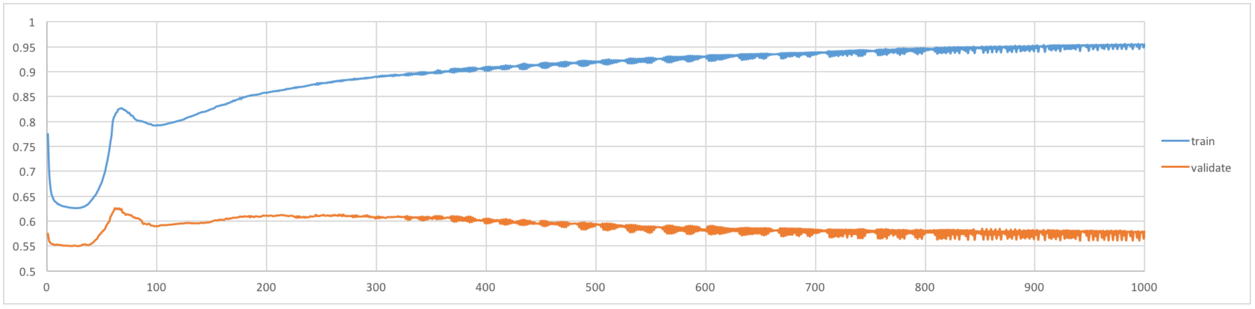
* "l2": Figure 6 shows the accuracy of training and validation for each feature set with "l2" regularization, over 1000 iterations. The precision, recall and f-score for training and validation set for each feature set is shown in Figure 7, 8, and 9, respectively. From Figure 6, 600 iterations for unigram and bigram is selected for testing and 400 iterations is selected for the one using both feature set. The test performance for unigram, bigram and both feature set is shown in Table 2.

2.2.3 Comparisons

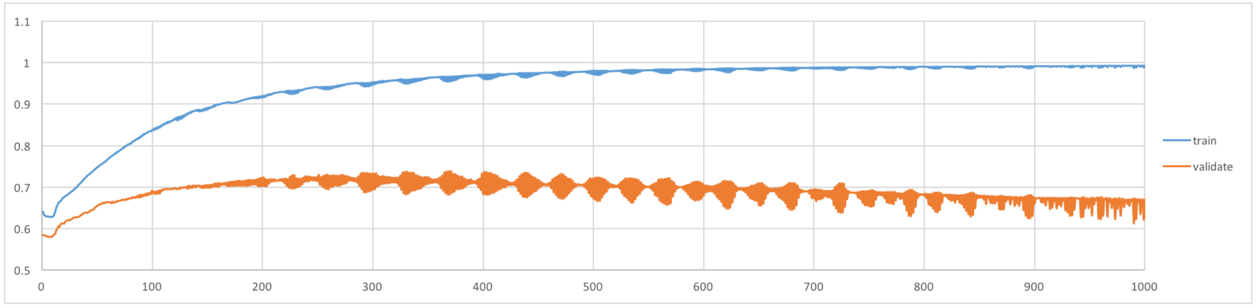
For both "I1" and "I2" norm, unigram feature set and using both feature set shows relatively higher accuracy than bigram feature set. When comparing between "I1" and "I2" norm, it can be observed that "I2" norm shows less ocsillation. Another important observation is that, "I2" norm is less likely overfitting than "I1" norm. Comparing Figure 2 and Figure 6, we can find "I1" norm validation accuracy decreases while the number of iterations increases. However, "I2" norm validation accuracy remains at a certain level while the number of iterations goes up.



(a)

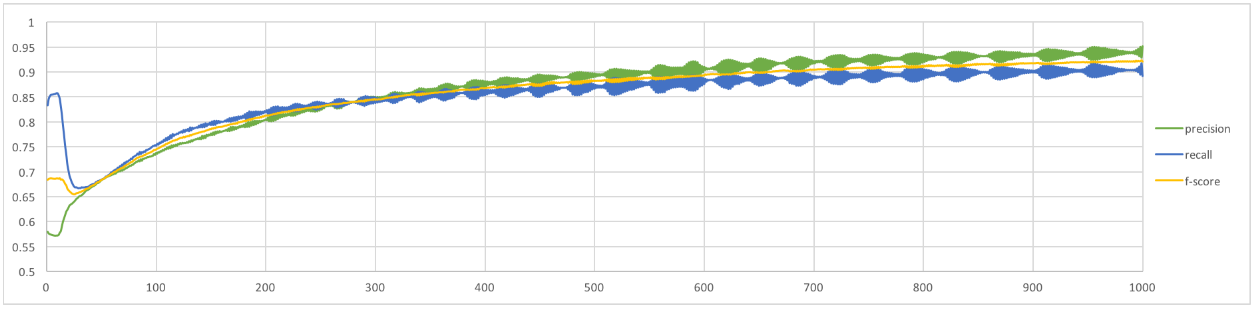


(b)

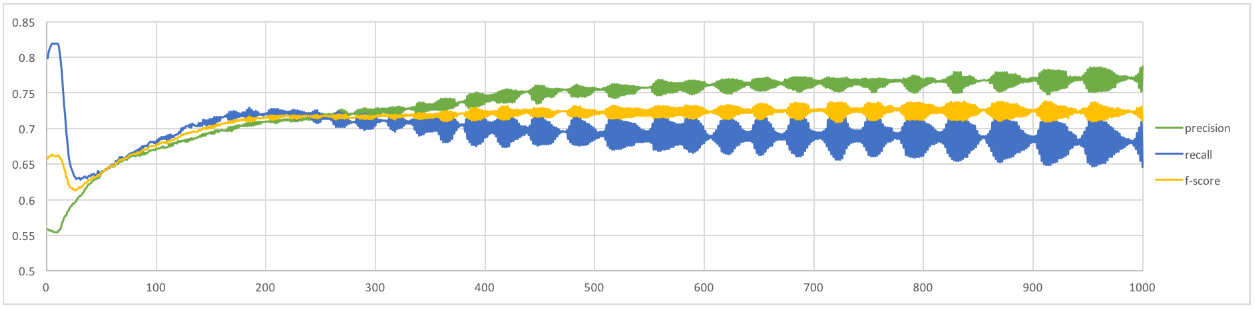


(c)

Figure 2: Accuracy of training and validation set for different feature sets with I1 norm.

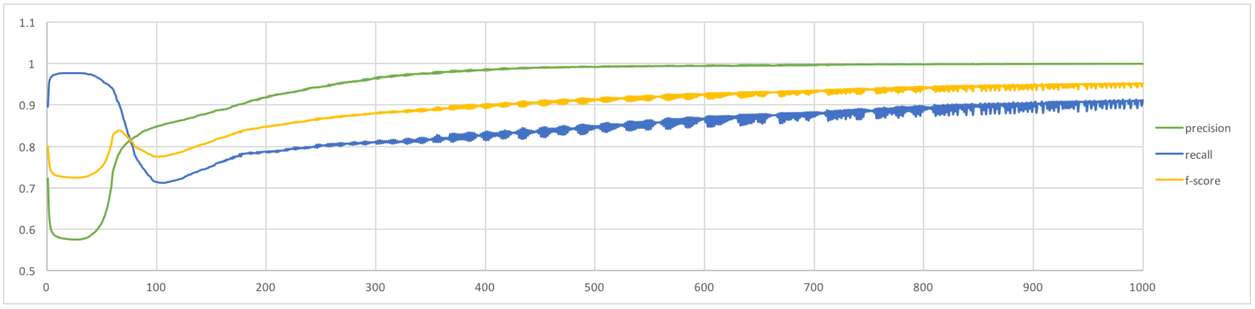


(a)

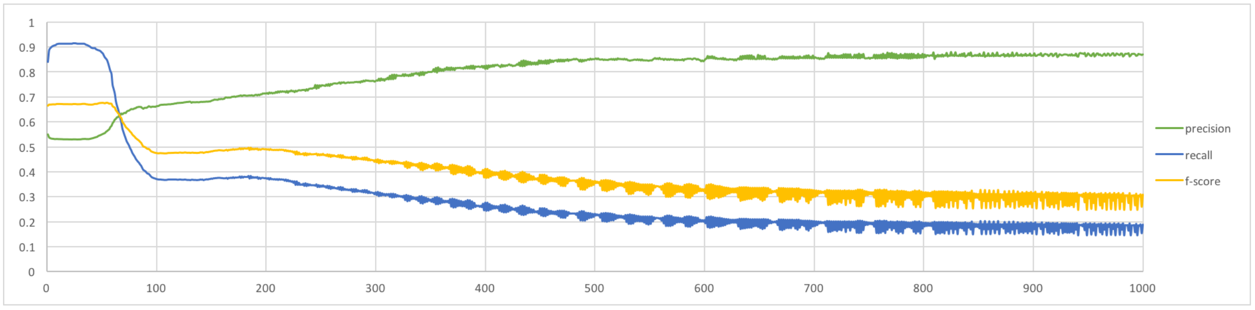


(b)

Figure 3: Performance of (a) training and (b) validation set for unigram feature sets with I1 norm.

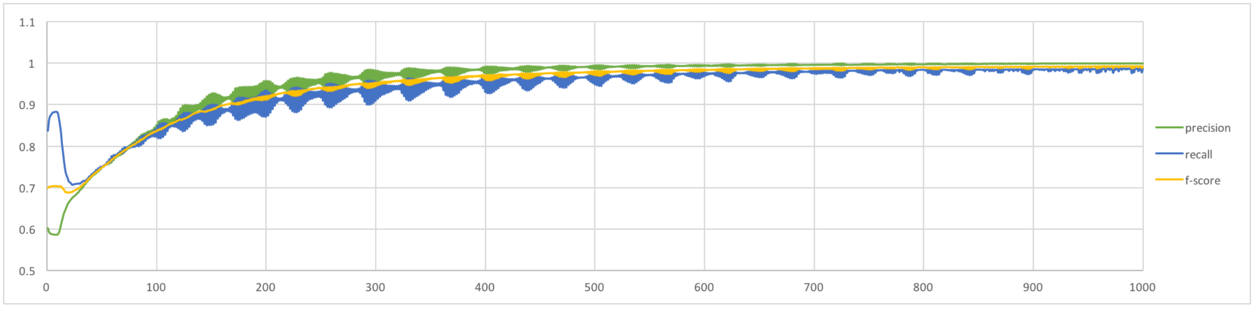


(a)

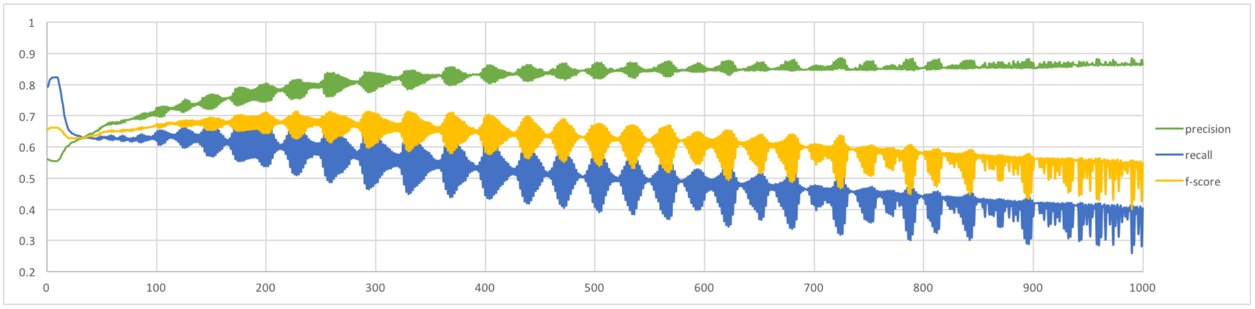


(b)

Figure 4: Performance of (a) training and (b) validation set for bigram feature sets with I1 norm.

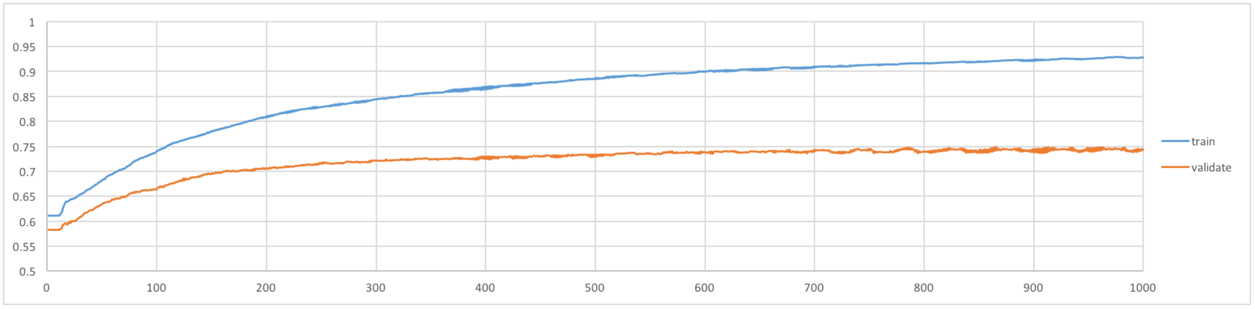


(a)

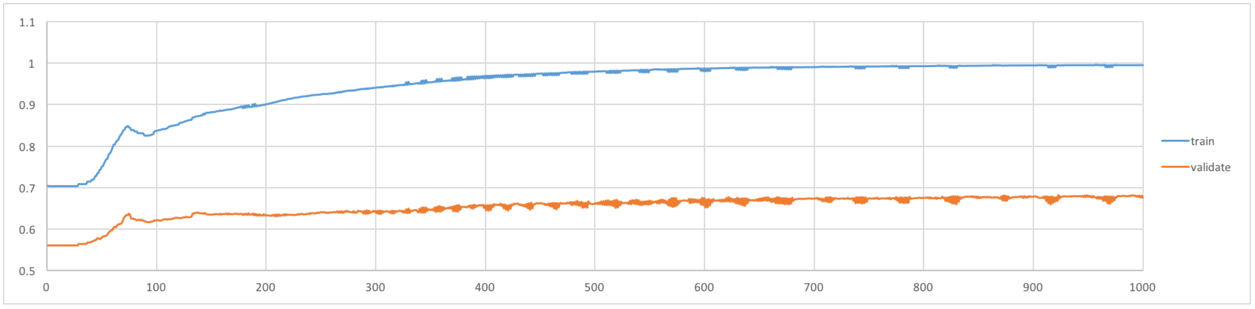


(b)

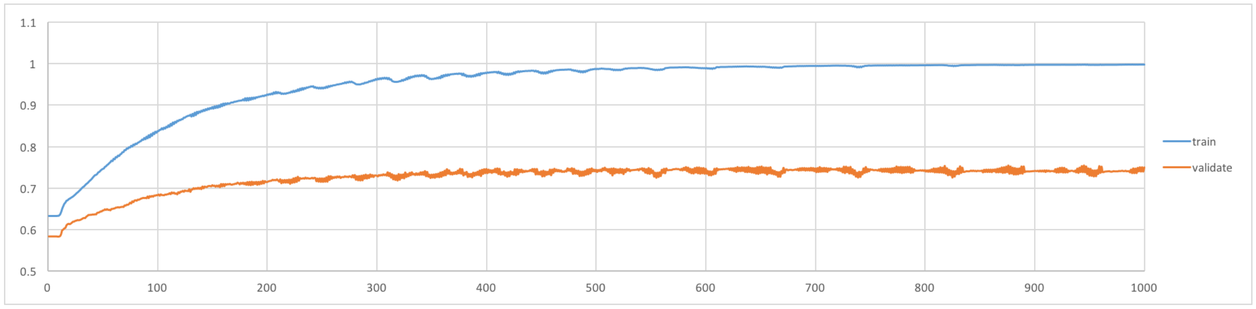
Figure 5: Performance of (a) training and (b) validation set for both feature sets with I1 norm.



(a)

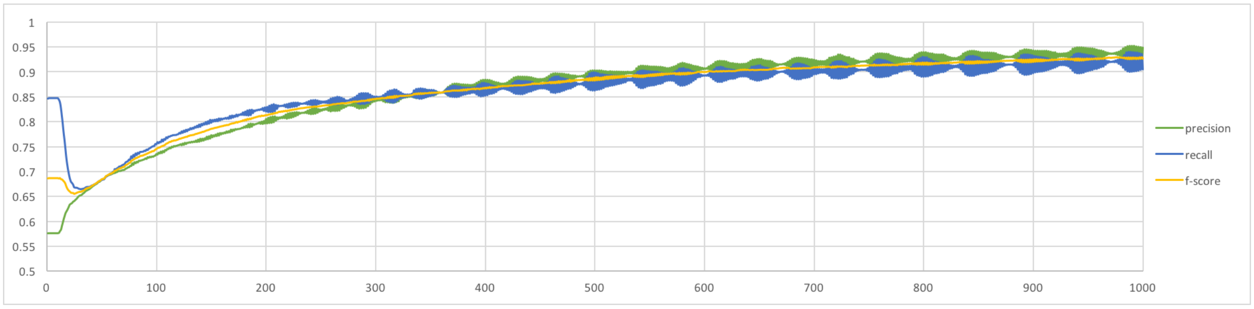


(b)

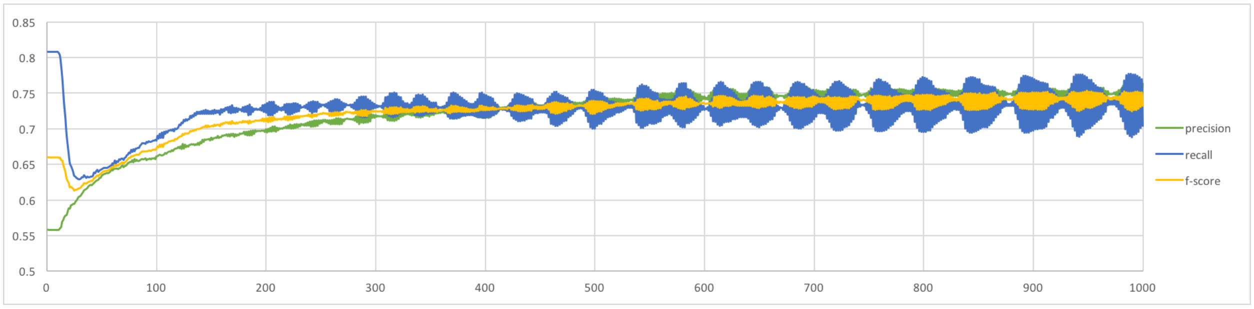


(c)

Figure 6: Accuracy of training and validation set for different feature sets with I2 norm.

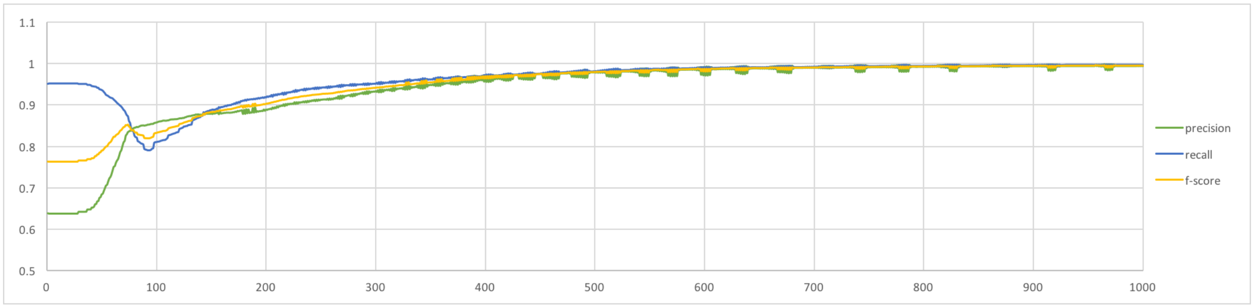


(a)

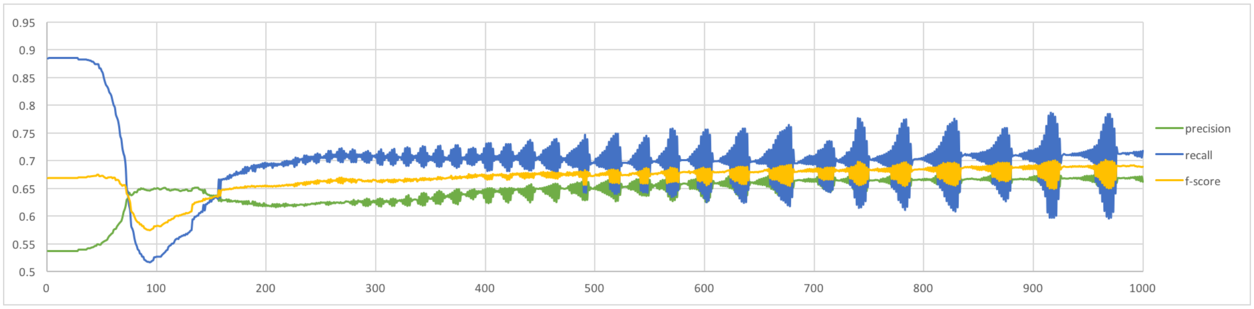


(b)

Figure 7: Performance of (a) training and (b) validation set for unigram feature sets with I2 norm.

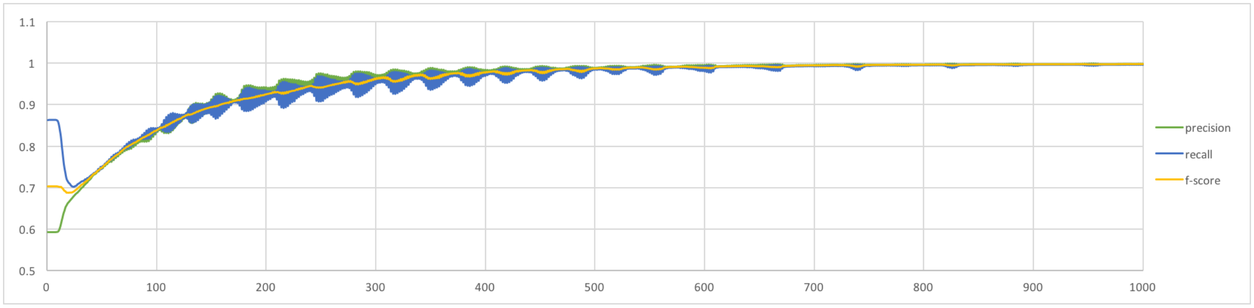


(a)

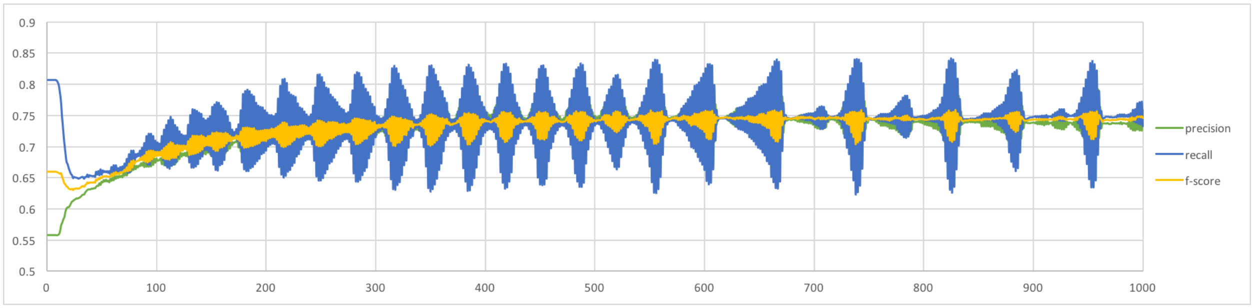


(b)

Figure 8: Performance of (a) training and (b) validation set for bigram feature sets with I2 norm.



(a)



(b)

Figure 9: Performance of (a) training and (b) validation set for both feature sets with l2 norm.