CSE 250B: Machine Learning, Fall '16

Homework 2: Sparse generative models

Gopal Rander (grander)

20 Newsgroup Data

60% train data (11269 documents), 40% test data (7505 documents)

Vocabulary (61066 words) after removing "stopword" (174 words)

List of "stopwords" used: https://pypi.python.org/pypi/stop-words

All the experiments are done after removing the "stopwords" in first step.

1. A short, high-level description of the idea for vocabulary selection

Intuition is to find words which are relevant to a class of documents. The relevance is determined by how frequent a word is in a class of documents and at the same time, how good can it distinctly represent that class.

For every class, we will select a word which satisfy these three properties and maximize the values of them.

1. Word frequency: The word should appear in the class more number of times compared to other words.
2. Class distinction: Distribution of number occurrences of the word should have a high variance over all the classes.
3. Class Weight: Given a word's occurrences has high variance over all classes, we weight the variance by percentage occurrences in each class.

For every word, we calculate the log frequency for each class. We also calculate the variance of every word's occurrences in each class weighted by the distribution percentage. To improve the classification, we also use the Inverse Document frequency heuristics.

1.) implies that we are selecting the most frequent word that appears in the training data for each class. It is highly likely that this word is observed in test data corresponding to the same class.

2.) implies that we are not selecting the words which are common in all the classes. For example, 'article' etc. Those words will not help in classifying a document distinctly. These words might occur a good number of times, so they got selected in step 1, but we need to filter them in step 2. We restrict ourselves to words with high frequency which also follow the high variance rule.

3.) Once we find a word with high variance, we choose the word if it identifies the class most likely, i.e. given the document with this word, the probability that the document belongs to the class should be high. For example, if the word occurs in some class >50% (given it passes through 1 & 2), then we are quite sure that this word distinctly identifies that class.

Summarizing the above 3 rules in formulas

For a class A_k ,we will select a word w based on following :-

Let w_{A_i} represent occurance of word w in class A_i

1. w has higher frequency than other words in class A_k.

$$Count(w_{A_k}) > Count\left(w'_{A_k}\right) \qquad \forall\, w' \in Vocab, \qquad w \neq w'$$

2. Variance of frequencies of w over all the classes is higher than that of other words over all the classes.

$$Var\left(Count(w_{A_1}), Count(w_{A_2}), \dots, Count(w_{A_k}), \dots\ Count(w_{A_n})\right) > Var\left(Count(w'_{A_1}), Count(w'_{A_2}), \dots, Count(w'_{A_k}), \dots\ Count(w'_{A_n})\right)$$

$$or\ VAR(w) > VAR(w')\ [short\ hand\ notation]$$

$$\forall\, w' \in Vocab, \qquad w \neq w'$$

3. Probability that w belongs to class A_k is higher than probability that w belongs to any other class. For probability, we use the maximum likelihood.

$$\Pr(w \in A_k^*) = \frac{Count(w_{A_k})}{\sum_i Count(w_{A_i})} > \frac{Count(w_{A_j})}{\sum_i Count(w_{A_i})} \qquad \forall\, i, j \in \{1,2 \dots n\}, \qquad j \neq k$$

A_k^* represents the set of words chosen to represent class A_k

Combining these three properties, we first calculate the distribution of words in each class, then we compute the $VAR(w)$ for each word. Then for each word, we calculate the prior probailty with respect to each class. Since we have to select words with highest values of all, we just multiply these values and select the word which has maximum value.

In matrix representation, let $MAT[A_i][w]$ represent position for word w and class A_i ($w \in Vocab,\ \ i \in \{1, 2, \dots n\}$)

Then, in step 1,

$MAT[A_i][w] = \log\left(\frac{Count(w_{A_i})+1}{\sum_{w'} Count(w'_{A_i})+size(vocab)}\right)$ (distribution of w in class A_i) We we take log to avoid underflow and also add smoothing factor.

In step 2, we multiply by the variance.

$MAT[A_i][w] = MAT[A_i][w] * VAR(w)$

Notice that variance is calculated with respect to words only. So every element in column w is multiplied by same term.

In step 3, we multiply by the word's class weight.

$MAT[A_i][w] = MAT[A_i][w] * \frac{1}{\sum_j MAT[A_j][w])}$

Then, for each class, we select the top p words which maximizes the value calcualted above. $p = \frac{M}{n\ (total\ number\ of\ classes)}$

$$A_i^* = \underset{w}{argmax_p}\, MAT[A_i][w] \qquad \forall\ i \in \{1, 2, \dots n\}$$

To improve it heuristically, we weight the word frequnecies with inverse document frequency (smoothened). This additional heuristics improves the class distinction for each word.

$$InverseDocFrequency(w) = \log\left(\frac{1}{1 + number\ of\ documnets\ containing\ w}\right)$$

Classification

Every test document t is represent in vector form using the subset of Vocabulary we got $t \equiv < COUNT(\mathrm{w}_1^{(\mathrm{t})}), COUNT(\mathrm{w}_2^{(\mathrm{t})}), \dots, COUNT(\mathrm{w}_{\mathrm{j}'}^{(\mathrm{t})}), \dots, COUNT(\mathrm{w}_{\mathrm{M}}^{(\mathrm{t})}) >$. Classification is done by selecting the class which maximizes $\Pr(A_i) * \Pi_{\mathrm{j}} \Pr(\mathrm{t} \in A_i | A_i)$. We take log to avoid underflow. In matrix format, we multiple the transpose of every class's vector with test document vector and multiply it with class prior probability. We label the document with the class which maximizes the above.

$$label(t) = \underset{\mathrm{i}}{\mathrm{agrmax}}(\Pr(A_i) * \Pi_{\mathrm{j}} \left(\mathrm{MAT}[A_i]\left[\mathrm{w}_{j'}^{(\mathrm{t})}\right]\right) * \mathrm{COUNT}(\mathrm{w}_{\mathrm{j}'}^{(\mathrm{t})}))$$

There might be cases where none of the words in the Vocabulary are found in the document. In that case, we do not use the above formula as we donot have enough information to classifiy. What we can do is to classify it to a label which is most likely i.e.

$$if\ t \equiv\ 0^{\rightarrow},\ \ label(t) = \underset{\mathrm{i}}{\mathrm{agrmax}}(\Pr(A_i))$$

2. Concise and unambiguous pseudocode

$$A_i \text{ represents a class. } w_j \text{ represent a word.}$$

$$n = \text{total number of classes,}$$

$$w_j, w_{j'} \in Vocabulary$$

Modified Selection (fullVocabulary, trainingSet, M)

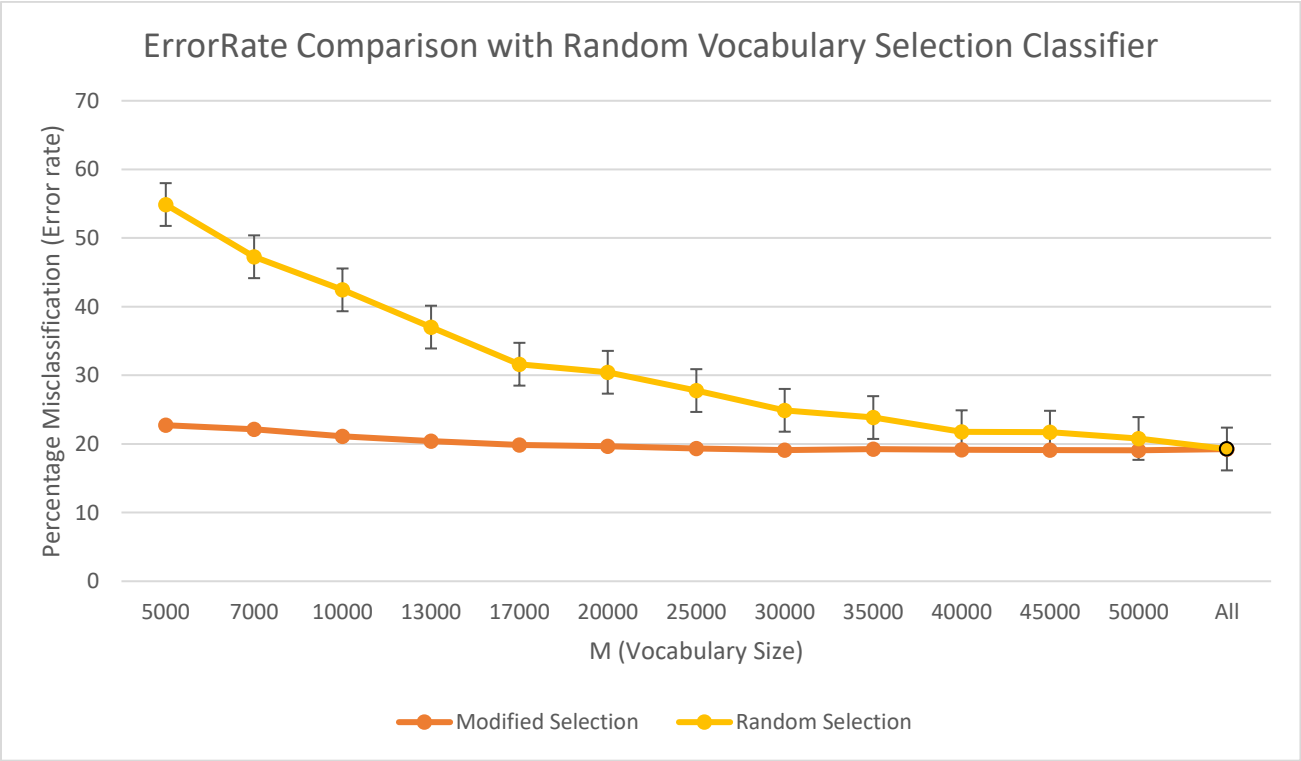
1. Remove the stopwords from Vocabulary
2. For each class, Calculate the class prior density. π_{A_i}
3. For each class, Calculate the probability distribution over the Vocabulary. Apply smoothing to the count of words and take natural log of it. $P_{A_i w_j}$
4. Calculate the inverse document frequency for each word in vocabulary. $\log(\frac{1}{idf_{w_j}})$ Weight the probability calculated in step 3 with inverse document frequency. $P_{A_i w} * log(\frac{1}{idf_{w_j}})$
5. Calculate the variance of every word's distribution across all classes. VAR_{w_j}
6. Calculate the class-wise distribution of word j. $1/\sum_i P_{A_i w_j}$
7. Weight $P_{A_i w_j}$ with VAR_w and $1/\sum_i P_{A_i w}$. $P_{A_i w_j} * VAR_{w_j} * 1/\sum_i P_{A_i w_j}$
8. For each class select top M/n words which have maximum value of $P_{A_i w_j}$ and keep them in another matrix $P^*_{A_i w_{j'}}$ Here j' represents the indices of words which we have selected.
9. ModifiedVocab = Reduced Vocabulary after selection. $\{w_{j'}\}$
10. Return ModifiedVocab

Classification(ModifiedVocab, P*, testSet, π)

1. Represent test document in terms of Modified vocabulary word counts. If test document does not contain any word from modified vocabulary, mark it invalid for now.
2. For valid documents, Use the classifier matrix $P^*_{A_i w_{j'}}$ to calculate $\Pi_i Count\left(w_{j'}^{(t)}\right) * P^*_{A_i w_{j'}}$ for each class. Multiply it with class prior density. π_{A_i}
3. Select the class which maximizes this value.
4. If the document was marked invalid, simply return the class which has the highest density.

3. Experimenatal Results

After removing the stopwords from vocabulary, We ran the Modfied Selection classifier model for various values of M ranging from 5000 to 60,000 and also the Random Selecrtion classifier for each M. The miscalssification percentage is summarized below in graph.



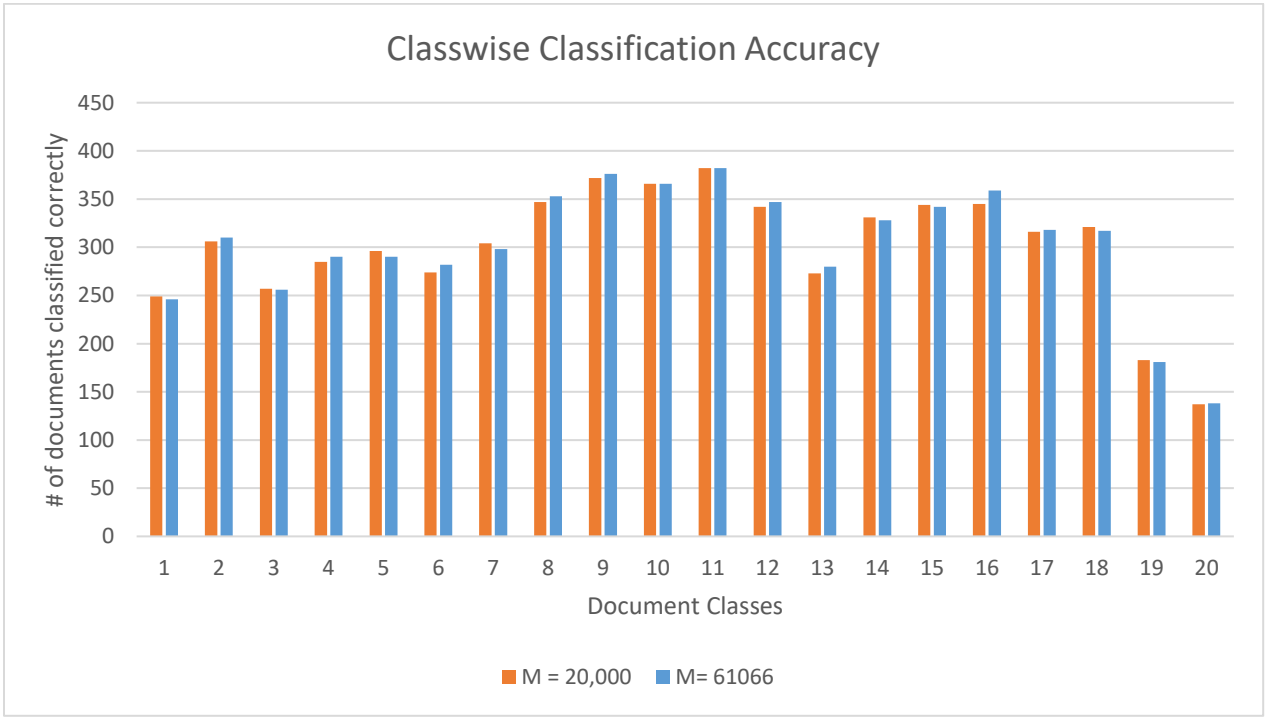
The bars in random classifier represent the error bars with upper limit and lower limit of standard error. For each value of M, the Random Vocbulary Selection Classifier was run 4 times.

We observe that the Modified Vocabulary Selection Classifier works much better for small values of M. Irrespective of values of M, the modified classifer is always better than the random classifer. Details of the graph are presented in the table below.

M	% Error Modified Selection	% Error Random Selection
5000	22.73151233	54.88341106
7000	22.11858761	47.2751499
10000	21.09260493	42.45169887
13000	20.42638241	37.02864757
17000	19.88007995	31.61892072
20000	19.65356429	30.44636909
25000	19.33377748	27.78147901
30000	19.10726183	24.90339773
35000	19.24050633	23.85076616
40000	19.17388408	21.78547635
45000	19.12058628	21.7188541
50000	19.08061292	20.79946702
All = 61066	19.26715523	19.26715523

Comparision with full vocabulary* (stopwords removed)

On the graph, the right-most value of M denotes the run for complete vocabulary and marks its error rate. The values are shown in the last row of the table above. We find that modified classifier is better than full vocab classifier if we choose ~25000 to ~30000 words.



4. Inspection of Models

Selecting class 'representatives'

For every class, after performing the rediction operation with M=1000, we sort the word according the relevance calculated above. The top twenty words for each class are :

1. Class : 'alt.atheism'
   Words : 'god', 'atheism', 'atheists', 'islam', 'religion', 'jesus', 'atheist', 'morality', 'bible', 'livesey', 'religious', 'islamic', 'moral', 'muslim', 'keith', 'belief', 'objective', 'evidence', 'argument', 'christian'

2. Class : 'comp.graphics'
   Words: 'graphics', 'image', 'jpeg', 'images', 'gif', 'format', 'animation', 'pu', 'ftp', 'files', 'algorithm', 'color', 'package', 'formats', 'op', 'amiga', 'processing', 'file', 'xv', 'vga'

3. Class : 'comp.os.ms-windows.misc'
   Words: 'windows', 'dos', 'ei', 'um', 'mouse', 'nt', 'files', 'font', 'file', 'win', 'di', 'card', 'microsoft', 'drivers', 'fonts', 'driver', 'swap', 'risc', 'printer', 'ql'

4. Class : 'comp.sys.ibm.pc.hardware'
   Words: 'scsi', 'ide', 'm', 'controller', 'drive', 'bios', 'drives', 'bus', 'isa', 'disk', 'card', 'dos', 'floppy', 'os', 'dx', 'irq', 'motherboard', 'vl', 'pc', 'dma'

5. Class : 'comp.sys.mac.hardware'
   Words: 'mac', 'apple', 'scsi', 'quadra', 'm', 'lc', 'simms', 'nubus', 'centris', 'fpu', 'duo', 'mhz', 'vram', 'monitor', 'ram', 'macs', 'simm', 'cpu', 'drive', 'modem',

6. Class : 'comp.windows.x'
   Words : 'window', 'widget', 'motif', 'entry', 'xterm', 'server', 'output', 'li', 'xli', 'contri', 'xt', 'file', 'export', 'printf', 'tar', 'char', 'application', 'null', 'display', 'entries'

7. Class : 'misc.forsale'
   Words : 'sale', 'shipping', 'dos', 'condition', 'manuals', 'disks', 'printer', 'stereo', 'brand', 'm', 'cd', 'floppy', 'meg', 'games', 'offers', 'manual', 'motherboard', 'rider', 'controller', 'modem'

8. Class: ‘rec.autos’
   Words: ‘car', ‘cars', ‘engine', ‘dealer', ‘autos', ‘oil', ‘tires', ‘honda', ‘mph', ‘ford', ‘toyota', ‘saturn', ‘wheel', ‘brake', ‘rear', ‘driving', ‘radar', ‘uoknor', ‘miles', ‘brakes'

9. Class: ‘rec.motorcycles’
   Words: ‘bike', ‘dod', ‘ride', ‘motorcycle', ‘bikes', ‘bmw', ‘riding', ‘helmet', ‘rider', ‘behanna', ‘honda', ‘mph', ‘rear', ‘nec', ‘c’, ‘rec', ‘cop', ‘tire', ‘gear', ‘engine'

10. Class : ‘rec.sport.baseball’
    Words: ‘baseball', ‘team', ‘season', ‘players', ‘braves', ‘cubs', ‘games', ‘game', ‘league', ‘teams', ‘player', ‘fans', ‘wins', ‘fan', ‘hr', ‘scored', ‘win', ‘philadelphia', ‘pitch', ‘stats'

11. Class : ‘rec.sport.hockey’
    Words: ‘hockey', ‘team', ‘nhl', ‘season', ‘players', ‘game', ‘leafs', ‘teams', ‘pts', ‘playoffs', ‘league', ‘rangers', ‘games', ‘detroit', ‘gm', ‘player', ‘wings', ‘det', ‘pittsburgh', ‘montreal'

12. Class : ‘sci.crypt’
    Words : ‘encryption', ‘clipper', ‘privacy', ‘escrow', ‘d’, ‘chip', ‘nsa', ‘keys', ‘security', ‘des', ‘secure', ‘algorithm', ‘government', ‘pgp', ‘enforcement', ‘eff', ‘anonymous', ‘agencies', ‘secret', ‘administration'

13. Class : ‘sci.electronics’
    Words : ‘circuit', ‘wiring', ‘wire', ‘voltage', ‘ground', ‘amp', ‘neutral', ‘detector', ‘electronics', ‘radar', ‘audio', ‘cooling', ‘mhz', ‘output', ‘wires', ‘chip', ‘connected', ‘input', ‘pin', ‘mc'

14. Class : ‘sci.med’
    Words : ‘msg', ‘patients', ‘disease', ‘health', ‘medical', ‘doctor', ‘food', ‘pitt', ‘treatment', ‘banks', ‘diseases', ‘surrender', ‘aids', ‘keyboard', ‘blood', ‘scientific', ‘studies', ‘drug', ‘water', ‘sci'

15. Class : ‘sci.space’
    Words : ‘space', ‘orbit', ‘launch', ‘nasa', ‘lunar', ‘shuttle', ‘moon', ‘satellite', ‘henry', ‘solar', ‘earth', ‘flight', ‘missions', ‘jpl', ‘rocket', ‘probe', ‘billion', ‘vehicle', ‘pat', ‘sci'

16. Class: ‘soc.religion.christian’
    Words: ‘god', ‘jesus', ‘christians', ‘christ', ‘church', ‘bible', ‘faith', ‘christian', ‘christianity', ‘rutgers', ‘scripture', ‘truth', ‘catholic', ‘sin', ‘lord', ‘heaven', ‘religion', ‘holy', ‘belief', ‘father'

17. Class : ‘talk.politics.guns’
    Words: ‘gun', ‘guns', ‘firearms', ‘weapons', ‘militia', ‘fbi', ‘batf', ‘weapon', ‘crime', ‘amendment', ‘police', ‘arms', ‘government', ‘criminals', ‘compound', ‘criminal', ‘stratus', ‘koresh', ‘waco', ‘tanks'

18. Class : ‘talk.politics.mideast’
    Words : ‘israel', ‘armenian', ‘turkish', ‘israeli', ‘armenians', ‘jews', ‘armenia', ‘turks', ‘ara’, ‘turkey', ‘jewish', ‘genocide', ‘greek', ‘soviet', ‘muslim', ‘killed', ‘peace', ‘civilians', ‘government', ‘war'

19. Class : ‘talk.politics.misc’
    Words : ‘stephanopoulos', ‘president', ‘government', ‘health', ‘drugs', ‘clinton', ‘tax', ‘gay', ‘myers', ‘br', ‘isc', ‘congress', ‘homosexual', ‘russia', ‘sexual', ‘insurance', ‘secretary', ‘administration', ‘senate', ‘billion'

20. Class : ‘talk.religion.misc’
    Words: ‘jesus', ‘god', ‘bible', ‘sandvik', ‘christians', ‘christian', ‘christ', ‘morality', ‘objective', ‘koresh', ‘christianity', ‘religion', ‘ra', ‘kent', ‘moral', ‘church', ‘newton', ‘greek', ‘jews', ‘faith'

    The selected words makes sense to me for each class. Also verified this with one friend of mine who guessed 18 correct classes out of twenty(Other two classes were closely related).

5. Critical Evaluation

The modified method of selecting a subset of vocabulary, size M for trianing is a clear win over the Random selection classifier with same vocabulary size. Moreover, our aim here was to reduce the vocabulary size drastically and still maintain a good classification accracy. Let’s comapre that. We know that taking all words of vocabulary (61066) gives an error rate of 19.27%, and a reduced vocabulary of size 5000 gives an error rate of 22.73%. In other words, reducing vocabulary size by 91.8% of its orginal size, we lose on only ~3.5% of accuracy. Also, we observed better accuarcy as compared to full vocabulary when we select only ~25000-30000 words.

M	Modified Vocabulary Accuracy
5000	77.26849
7000	77.88141
10000	78.9074
13000	79.57362
17000	80.11992
20000	80.34644
25000	80.66622
30000	80.89274
35000	80.75949
40000	80.82612
45000	80.87941
50000	80.91939
61066	80.73284

Scope for improvement

Considering Covariance

In the modified selection method, we have used multinomial Naïve Bayes classifier which has a assumption that all the features are independent. But as we know and it is quite evident that words in document are not independent of each other. And this information can help us in reducing the feature vector by combining the words whose occurances have high co-variances. These words in one bag will contribute to a single parameter.

For example, words like 'god' and 'religion' are in the top representatives of every class related to religion. We can find such words after vocabulary reduction as finding co-variance among 60,000 words seems infeasible, but for 10000, it is bit feasible. Another solution might be considering covariance in bags of words before reducing the vobaulary size.

Adding More training data

Since we had around 11,000 documets to train our model and then tested it on around 7500 documents, what I feel is we can increase the training set data for better classification. Although it might not help in reducing the feature set i.e. the vocabulary, but it can help in better overall classification. For example, when we took 5000 words for vocabulary, 7 documents in test set did not have any word out of them, so we decided on

the basis of each class's prior probability. If we have more training data, we can minimize this case so that at least one word of the subset is found in test data.

Other ML hueristics & techniques for classification

Other hueristic techniques on word selection might be applied. Some of them are:

1. χ^2 based feature selection
2. Greedy approach for incremental feature selection.

Other techniques like SVM based models are helpful and efficient in case of large number of features as the classification depends only on a small subset of features.

Try Next

On the similar lines of scope for improvement, I would like to try out other models like SVM for document classification. I would also like to check any subset of modified vocabulary gives similar or better classification accuracy. In that case, I will try a decremental approach to find a smallest subset to represent each class.

Most of the misclassification happens on the boundaries, i.e. between two classes which are closely related. In such cases, Linear regresssion can be used to find the optimal classification.