# Classifier ensemble for biomedical document retrieval

**Manabu Torii[1§], Hongfang Liu[1]**

[1]Department of Biostatistics, Bioinformatics, and Biomathematics, Georgetown University Medical Center, 4000 Reservoir Rd, NW, Washington, DC 20057, USA

[§]Corresponding author

Email addresses:

MT: mt352@georgetown.edu

HL: hl224@georgetown.edu

# Abstract

## Background

Due to rich information embedded in published articles, literature review has become an important aspect of research activities in the biomedical domain. Machine Learning (ML) techniques have been explored to retrieve relevant articles from a large literature archive (i.e., classifying articles into relevant and irrelevant classes), and to accelerating the literature review process. Meanwhile, an ensemble classifier, a system that assigns classes based on the outputs of multiple classifiers, tends to be more robust and has better performance than each individual classifier. Ensemble classifiers are often composed of classifiers trained on different training sets (e.g., sampled data sets) or of those using different ML algorithms. In this paper, we propose a simple ensemble approach where an ensemble is composed of classifiers using different feature sets for an ML algorithm. We evaluated the approach using Support Vector Machine (SVM) on two publicly available collections of MEDLINE citations, the Post-translational modification (PTM) data sets and the Immune Epitope Database (IEDB) data sets, that resulted from biomedical database curation projects.

## Results

The evaluation showed that ensemble classifiers outperformed their constituent classifiers as measured by both area under ROC curve (AUC) and precision/recall break-even-point (BEP), provided with enough training data. We observed that the performance of SVM ensembles were competitive or better than the best results previously reported for the data sets used.

## Conclusions

The proposed ensemble approach was found to be effective in improving performance of SVM classifiers. The approach is also simple and easy-to-deploy in document classification/retrieval tasks. However, improvement of classifiers through the current approach is still modest. We plan to explore different ways to derive and combine constituent classifiers, and continue our investigation over other data sets.

## Background

Due to rich information embedded in published biomedical articles, literature review has become an increasingly important aspect of research activities in the biomedical domain, e.g., [1, 2]. One of the initial steps in literature review is document retrieval, i.e., gathering documents relevant to the target topic from a large literature archive such as MEDLINE. To mitigate human effort in retrieving relevant articles, there has been a growing interest in automatic document retrieval. It has become an active research area in biomedical text mining. For example, Genomics Track (2003–2007) of Text Retrieval Conference (TREC) has dedicated to the evaluation of biomedical document retrieval systems [3]. Also, one of the tasks in BioCreAtIvE II[1] (http://biocreative.sourceforge.net/biocreative_2.html) asked participants to order biomedical articles based on their relevance to protein interaction annotation.

Document retrieval is to prioritize (i.e., order) documents according to their relevance to the target topic. Considering the target documents as positive instances and others as negative instances, the priority order of documents can be obtained using a classifier that yields a confidence score in assigning positive/negative classes to documents. Machine Learning (ML) approaches such as Naïve Bayes and Support Vector Machine (SVM) have enjoyed great success in document classification and retrieval [4]. In the biomedical domain, ML classifiers have been considered for document retrieval in database curation projects [5-9], and therefore the efficiency of database curation can be enhanced by improving document classifiers. For a specific application, improvement of ML classifiers can be attempted through incorporation of task-specific features/heuristics and/or elaborated domain-specific features [3, 7, 8]. Alternatively, or in conjunction with such effort, classifier performance can be improved by combining multiple classifiers, i.e., ensemble of classifiers [10].

In this study, we consider a simple and easy-to-deploy ensemble approach for biomedical document classification tasks where an ensemble is composed of classifiers built

[1] Protein Interaction Article Sub-task (IAS) of the Protein-Protein Interaction task (PPI).

with different sets of features. The goal of this study is two fold: i) examine the effectiveness of our ensemble approach for classification of MEDLINE citations; and ii) report classification performance on publicly available data sets in the domain.

In the following, we first provide background information for classifier ensemble. Next, we describe two publicly available data sets used in this study, the Post-translational modification (PTM) data sets [8] and the Immune Epitope Database (IEDB) data sets [7], resulted from actual biomedical database curation projects.

## Classifier ensemble

It has been observed that "accurate and diverse" classifiers make an ensemble classifier that outperforms a single classifier [10, 11]. A classifier is "accurate" if it performs better than a random classifier and classifiers are "diverse" if they do not make the same classification mistakes. Popular ensemble approaches include bagging and boosting. In the bagging approach [12], constituent classifiers of an ensemble are trained on data sets sampled from the training data. In the boosting approach [13], constituent classifiers are trained sequentially, in which misclassified instances are assigned more weights during the training of the next classifier. The performance of bagging and boosting is dependent on the ML algorithm used. For example, in the newswire domain, Dong and Han [14] examined the utility of different ensemble methods including bagging and boosting for document classification. In their work, although a boosted Naïve Bayes classifier outperformed a single Naïve Bayes classifier, a boosted SVM classifier performed worse than a single SVM classifier. In fact, since SVM does not depend on weights/frequencies of instances, boosting may not be an appropriate choice for SVM (see, e.g., [15]). In Dong and Han, there was little or no improvement reported also for bagging with Naïve Bayes and with SVM.

In bagging or boosting, constituent classifiers of an ensemble are built by varying training data sets (i.e., using sampled documents or documents with different weights). In this study, we propose an ensemble approach that builds constituent classifiers by varying the size of feature words used (i.e., by varying feature vectors, but using the same document set and the same ML algorithm).

## Publicly available data sets for biomedical document classification/retrieval

**PTM data sets**[2] – The PTM data sets developed at Protein Information Resource (PIR) consist of five collections of MEDLINE citations for five different PTM types (acetylation, glycosylation, hidroxylation, methylation, and phosphorylation), which were labelled by domain experts as either positive or negative at the level of abstract or full-length article. Small parts of the abstract-level PTM data sets have been used in a document retrieval study by Han et al. [8], which specifically investigated document retrieval for small data sets. We used these small data sets for performance comparison purposes. Of five data sets used in Han et al., we used two data sets, the acetylation and phosphorylation data sets, where there are at least 50 positive documents[3] (Table 1). In the work reported in [8], Naïve Bayes classifiers exploiting substring features outperformed SVM classifiers on these data sets.

**IEDB data sets** [7][4] – The IEDB data sets were developed during the annotation of epitopes from four different sources with a view to populating the Immune Epitope Database. Thus, the data sets consist of four sets of MEDLINE citations (abstracts and titles), which were retrieved from PubMed using "complex queries." Citations were, then, manually classified as positive and negative documents. In this study, following the study of Wang et al. [7], we combined the four data sets and derived a corpus of 20,907 MEDLINE citations. In [7], the authors reported that Naïve Bayes classifiers outperformed SVM classifiers in the

[2] The PTM data sets are available at the PIR iProLink web site (http://pir.georgetown.edu/cgi-bin/ipkLitFt.pl?stat=12). In this study, we used smaller parts of these data sets, which were introduced in the study by Han et al. (http://www.ist.temple.edu/PIRsupplement/).

[3] The acetylation data set contains 55 references to MEDLINE citations (PMIDs) for positive documents. However, when the citations were downloaded, six of them were no longer accessible with the listed PMIDs (see the caption of Table 1).

[4] http://www.biomedcentral.com/1471-2105/8/269/additional/

experiments, and the best classification performance was obtained using domain- and task-specific features. Summaries of the data sets are found in Table 1.

# Methods

We used SVM Light [16] to derive SVM classifiers in this study. Specifically, we used radial basis function (RBF) with gamma value of 1.0 as the kernel, with the default setting for the regularization parameter C in SVM Light. We used this setting for its yielding the good performance in preliminary experiments using the small portion of the data sets. For classification features, only normalized words in documents are used without any task- or domain-specific features/heuristics.

## Word normalization

For SVM classifiers, we used words in documents as features, except for stop words listed in the NCBI stopword list[5] and rare words that appear in less than three documents in a training data set. Words are defined as consecutive alphabet letters, numbers, hyphens (-), or slash (/), e.g., IL-1 is regarded as one word without being tokenized into smaller sequences. All words were processed with our implementation of S-stemmer [17], which converts plural nouns and third person singular verbs into their base forms, e.g., receptors → receptor, studies → study, IFNs → IFN. Also, within each word, alphabet letters were lowercased, a number sequence was converted to a token "DIGIT", and Greek alphabets to a token "GREEK", e.g., KappaB → GREEKb and Ras1 → rasDIGIT.

## Feature vector generation

To select classification features among normalized words identified in a training data set, we used information gain (IG), also known as expected mutual information, commonly used in text classification, e.g., [7, 8, 18, 19]. IG is calculated:

$$IG(D,w)=H(D)-\sum_{d\in D^{w+}}\frac{\left|D^{w+}\right|}{\left|D\right|}H\left(D^{w+}\right)-\sum_{d\in D^{w-}}\frac{\left|D^{w-}\right|}{\left|D\right|}H\left(D^{w-}\right)$$

[5] http://www.ncbi.nlm.nih.gov/books/bv.fcgi?indexed=google&rid=helppubmed.table.pubmedhelp.T43

where $H(\bullet)$ is an entropy for having different classes given a document set, and $D^{w+}$ and $D^{w-}$ are partitions of a document set $D$, each of which consists of documents containing ($w+$) or not containing ($w$-) word $w$, respectively. In building classifiers, we consider the top R percent of words according to the IG measure. In this study, ten different R values were considered: R = 1, 4, 9, …, 100, i.e., $R=n^2$ for n = 1, 2, ..., 10. Among the words with higher IG values, differences of IG values for any two words tend to be greater, while among those with lower IG values, such differences tend to be smaller[6]. This motivated us to use the above choices of R in deriving a set of "diverse" classifiers. Namely, we varied the percentage less for the smaller values of R, e.g., 1 → 4 → 9 → …, but varied the percentage more for the larger values of R, e.g., … → 64 → 81 → 100.

A document was represented with a set of selected feature words found therein (a bag-of-words approach) in a vector format. Each value in a vector associated with a feature word is a frequency of words (i.e., terms) in the document (TF) weighted by the inverse of the document frequency (IDF), i.e., $TF_{i,j} \times \log(IDF_j)$ for word $j$ in document $i$. As in [19], feature vectors are normalized so that the Euclidean norm of a vector is 1.0.

### Classifier ensemble

Given a set of input documents, a classifier will assign a numeric value to each document, which is regarded as a confidence score for a positive (or negative) class. With a single classifier, documents are ranked according to assigned confidence scores. With multiple classifiers, documents can be raked according to the summation of confidence scores assigned to each document.

[6] To be clear, let $w_1, w_2, w_3, \ldots, w_n$ be an ordered list of all words in a corpus such that $IG(w_1) \geq IG(w_2) \geq \ldots \geq IG(w_n)$, where $IG(\bullet)$ is a function mapping a word to an IG value. Then, roughly speaking, we observed $IG(w_i)-IG(w_{i+1}) > IG(w_{i+1})-IG(w_{i+2})$ for $i=1\ldots n-1$. In other words, comparatively speaking, an SVM classifier exploiting 1% of top IG-value words, say $SVM_{R=1}$, can differ much from another classifier $SVM_{R=4}$, while $SVM_{R=97}$ and $SVM_{R=100}$ may be almost the same in terms of their performance.

In order to derive multiple classifiers from one training data set, we built each classifier by varying the threshold value R in selecting features. As detailed in the previous sub-section, a set of ten classifiers are derived using ten different R values (i.e., R=1, 4, 9, 16, …, 100). Among these single classifiers, a group of two (R=1 and 4), three (R=1, 4, and 9), four (R=1, 4, 9, and 16), …, ten (R=1, 4, 9, … 100) classifiers were selected so that each group of classifiers makes an ensemble classifier, i.e., nine ensemble classifiers.

## Classifier evaluation

Two measures were used to evaluate the performance of classifiers: Area under ROC curve (AUC) and Precision/recall break-even-point (BEP). Given an ordering of documents by a classifier, AUC is interpreted as the probability that the rank of a positive document $d_1$ is greater (i.e., more likely to be positive) than that of a negative document $d_0$, where $d_1$ and $d_0$ are documents randomly selected from positive and negative document sets, respectively. The higher the AUC value, the better the classifier is. After documents (of size n) are ranked from 1 (least likely to be positive) to n (most likely to be positive), AUC can be calculated as $AUC = \frac{S - n_1(n_1 + 1)/2}{n_0 n_1}$, where $S$ is the sum of the ranks assigned to positive documents, and $n_0$ and $n_1$ are the numbers of negative and positive documents, respectively (see the details in [20]). BEP is a precision (or a recall)[7] obtained for a classification threshold where the precision and the recall become equal (see, e.g., [19]).

While AUC can provide a summary of the overall ordering of positive and negative documents by a classifier, when comparing classifiers from literature reviewers' point of view, a higher AUC value does not necessarily imply the better utility of the classifier. For example, suppose reviewers can go over at most 100 articles among the list of articles

[7] For a fixed threshold on confidence scores assigned by a classifier, documents are classified into two classes. Then, precision is the number of true positives divided by the total number of true positives and false positives. Recall is the number of true positives divided by the total number of positive instances.

retrieved at a time. Then, for reviewers, changes in document ordering matter only when they take places within the first 100 documents. In this respect, BEP may be the more appropriate performance measure.

Each classifier was evaluated in m repeated *n*-fold cross-validation, and average AUC and BEP over $m \times n$ runs were calculated, i.e., a document collection was split into *n* equally-sized partitions, and classifiers trained on *n-1* partitions were evaluated over the remaining one partition, which was repeated *n* times using a different partition as a test set each time. Then, such n-fold cross-validation test was repeated *m* times over the data collection. For each data set, the same partitioning was used for all the evaluated classifiers. To make the results comparable to the previously reported results, different pairs of *m* and *n* were used for the data sets. ($m$=20 and $n$=5 for the PTM data sets, and $m$=1 and $n$=10 for the IEDB data sets).

# Results and discussion

## Results on the PTM data sets

On the two PTM data sets (acetylation and phosphorylation), we evaluated ten single SVM classifiers and nine ensemble classifiers as detailed in the Method section. For each single and ensemble classifier, we repeated 5-fold cross-validation tests 20 times as in [8], and calculated average AUC and BEP measures of, thus, 100 runs. In each run, there were about 2,500 and 1,800 unique words in the training set portion of the acetylation and phosphorylation data sets, respectively. R% of the unique words were used in training a single classifier. The results of the experiments are reported in Table 2.

We found inclusion of excessive word features was harmful for these small data sets, although SVM can usually exploit a large number of word features including those that are less informative (e.g., in terms of IG) [19]. For both of the data sets, the best performance of single SVM classifiers was obtained at R = 4 in term of AUC measure, and R = 9 in terms of BEP (Table 2). The best performance of ensemble classifiers was obtained when five single classifiers (R=1, 4, 9, 16 and 25) were combined for the acetylation data set, and when four classifiers (R=1, 4, 9 and 16) were combined for the phosphorylation data set. For both of the data sets, performance of ensemble classifiers was consistently better than single classifiers in terms of both AUC and BEP.

We found the performance of single SVM classifiers and that of the most comparable SVM classifier in [8] (WB-SVM-IG) were still very different. While further investigation is needed, this difference may be attributed to different classifier settings (e.g., a linear kernel function in [8] vs. an RBF kernel function in our experiments for SVM), document representation (e.g., we used normalized TF-IDF vectors in our experiments, while it is not clear in [8]), and/or threshold settings in feature selection (i.e., a fixed threshold, IG > 0.02, in [8]).

We also examined the applicability of the ensemble approach on the glycosylation, hydroxylation and methylation data sets used in Han et al, where there are very small numbers of positive instances (Table 1). On these data sets, performance of ensemble classifiers was no better than that of single classifiers, or sometimes even worse. On the hydroxylation and methylation data sets where there are especially small numbers of positive and negative documents, BEP of single classifiers were low (e.g., an average BEP of ten single classifiers was 0.24 and 0.47 for the hydroxylation and the methylation data set, respectively). We assumed that such classifiers were not reliable enough to contribute to an ensemble classifier.

## Results on the IEDB data sets

On the combined IEDB data sets, ten single SVM classifiers and nine ensemble classifiers were evaluated just like on the PTM data sets. Compared to the PTM data sets used by Han et al., there are a much larger number of documents in this data set (20,907 MEDLINE citations as opposed to 916 or 457 citations), and we obtained stable results in a ten-fold cross-validation test. In each fold, there were about 22,000 unique words in the training set portion of the data set, R% of which were used as features in training a single classifier. The results are shown in Table 3. Figure 1 shows how AUC and BEP change as R changes for single and ensemble classifiers. Note that, for ensemble classifiers, R is to indicate the largest percentage of feature words used among constituent classifiers, e.g., an ensemble classifier consists of single classifiers using 1, 4, 9, …, up to R% of word features.

As in Figure 1, for single classifiers, the BEP measure peaks at R=16 and it degrades when R < 16 or R > 16. On the other hand, performance of ensemble classifiers keeps improving as R gets larger. The ensemble classifier with R=100 outperformed all the single SVM classifiers in terms of both AUC and BEP (Table 2).

To examine the applicability of this ensemble approach to Naïve Bayes methods, we repeated the same experiment using the MALLET library [21] to build multinomial Naïve Bayes classifiers. The results are reported in Table 3 and Figure 2. Table 3 shows that performance (i.e., AUC) of Naïve Bayes classifiers in this study agrees with that in [7] (despite that [7] used binary feature vectors and we used TF feature vectors, see, e.g., [18]). As in Table 3 and Figure 2, although the proposed ensemble approach improved classification performance of Naïve Bayes classifiers in terms of AUC, it did not improve in terms of BEP.

While these results need to be confirmed on other data sets, the success of the proposed ensemble approach may be attributed to the property of SVM classifiers that they can exploit a large number of features (even less informative features in terms of IG) with hardly over-fitting to data sets [19]. Namely, given a larger number of words as features, SVM classifiers will yield a globally well-ordered document list without over-fitting. On the other hand, given a small number of the top IG value words, SVM classifiers will identify apparently positive and apparently negative documents confidently. Thus, the ensemble classifiers will take advantage of the both ranking schemes. This did not hold for Naïve Bayes classifiers, whose performance (BEP) degraded when a large number of features were used.

## Conclusions

In this study, we examined a simple and easy-to-deploy classifier ensemble approach for biomedical document classification/retrieval tasks. In the proposed approach, constituent classifiers were built by varying the sizes of the feature set for an ML algorithm. Note that even when a single classifier is employed in a database curation project, a number of classifiers with different sizes of feature sets would be built anyway before the best performing system is selected. The proposed approach suggests combining such intermediate classifiers. In our experiments, SVM ensembles outperformed all the constituent classifiers in terms of both AUC and BEP. Using this approach, we updated the classification performance previously reported on the benchmarking data sets, and set new baseline performance for the data sets. However, the ensemble approach was not effective when there was no sufficient data to train reliable constituent classifiers or when it was applied to Naïve Bayes classifiers.

In the current ensemble method, the way we derive constituent classifiers is based on our observation of the list of feature words. We plan to explore systematic ways in selecting different sets of features, and different approach to combining resulted classifiers. We also plan to investigate the effectiveness of the method using different data sets and different ML algorithms.

## Authors' contributions

MT carried out the experiments and drafted the manuscript. HL participated in the design of the study and helped draft and revise the manuscript. MT and HL both read and approved the final manuscript.

## Acknowledgements

We thank Zhangzhi Hu at PIR as well as anonymous reviewers of LBM for helpful comments on the manuscript. We also thank those researchers who made the corpora and the machine learning software publicly available. This project was supported by IIS-0639062 from the National Science Foundation.

## References

1. Farriol-Mathis N, Garavelli JS, Boeckmann B, Duvaud S, Gasteiger E, Gateau A, Veuthey AL, Bairoch A: **Annotation of post-translational modifications in the Swiss-Prot knowledge base**. *Proteomics* 2004, **4**(6):1537-1550.
2. Peri S, Navarro JD, Kristiansen TZ, Amanchy R, Surendranath V, Muthusamy B, Gandhi TK, Chandrika KN, Deshpande N, Suresh S *et al*: **Human protein reference database as a discovery resource for proteomics**. *Nucleic Acids Res* 2004, **32**(Database issue):D497-501.
3. Hersh W, Bhupatiraju RT, Corley S: **Enhancing access to the Bibliome: the TREC Genomics Track**. *Medinfo* 2004, **11**(Pt 2):773-777.
4. Sebastiani F: **Machine learning in automated text categorization**. *Acm Computing Surveys* 2002, **34**(1):1-47.
5. Donaldson I, Martin J, de Bruijn B, Wolting C, Lay V, Tuekam B, Zhang S, Baskin B, Bader GD, Michalickova K *et al*: **PreBIND and Textomy--mining the biomedical literature for protein-protein interactions using a support vector machine**. *BMC Bioinformatics* 2003, **4**:11.
6. Dobrokhotov PB, Goutte C, Veuthey AL, Gaussier E: **Combining NLP and probabilistic categorisation for document and term selection for Swiss-Prot medical annotation**. *Bioinformatics* 2003, **19 Suppl 1**:i91-94.
7. Wang P, Morgan AA, Zhang Q, Sette A, Peters B: **Automating document classification for the Immune Epitope Database**. *BMC Bioinformatics* 2007, **8**(1):269.
8. Han B, Obradovic Z, Hu ZZ, Wu CH, Vucetic S: **Substring selection for biomedical document classification**. *Bioinformatics* 2006, **22**(17):2136-2142.
9. Shah PK, Bork P: **LSAT: learning about alternative transcripts in MEDLINE**. *Bioinformatics* 2006, **22**(7):857-865.
10. Dietterich TG: **Ensemble methods in machine learning**. *Multiple Classifier Systems* 2000, **1857**:1-15.

11. Chung YS, Hsu DF, Tang CY: **On the Diversity-Performance Relationship for Majority Voting in Classifier Ensembles**. In: *7th International Workshop on Multiple Classifier Systems (MCS) 2007*; Springer Verlag.
12. Breiman L: **Bagging predictors**. *Machine Learning* 1996, **24**(2):123-140.
13. Freund Y: **Boosting a Weak Learning Algorithm by Majority**. *Information and Computation* 1995, **121**(2):256-285.
14. Dong Y, Han K: **A Comparison of Several Ensemble Methods for Text Categorization**. In: *Services Computing, 2004 IEEE International Conference on (SCC) 2004*: pp. 419-422.
15. Kudo T, Matsumoto Y: **Chunking with Support Vector Machines**. In: *The Second Meeting of North American Chapter of Association for Computational Linguistics (NAACL)* 2001: pp.192-199.
16. Joachims T: **Making large-Scale SVM Learning Practical**: MIT-Press; 1999.
17. Harman D: **How effective is suffixing?** *Journal of the American Society for Information Science* 1991, **42**(1):7-15.
18. McCallum AK, Nigam K: **A Comparison of Event Models for Naive Bayes Text Classification**. In: *AAAI/ICML-98 Workshop on Learning for Text Categorization*: AAAI Press; 1998: pp. 41-48.
19. Joachims T: **Text Categorization with Support Vector Machines: Learning with Many Relevant Features**. University of Dortmund; 1997.
20. Hand DJ, Till RJ: **A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems**. *Machine Learning* 2001, **45**(2):171 - 186.
21. McCallum AK: **MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu**; 2002.

## Table 1 - Summaries of PTM and IEDB data sets

The parts of the PTM data sets used in Han et al. are available as lists of PubMed Unique Identifiers (PMIDs) assigned to MEDLINE citations. Some of the citations, however, are no longer accessible with the listed PMIDs. In this table, numbers in parentheses are the document counts reported in the original papers introducing the data sets. We used two of the five data sets from Han et al., the acetylation and phosphorylation data sets (indicated by * in the table), where there are (originally) more than 50 positive documents.

| Data sets | | Positives | Negatives |
|---|---|---|---|
| Data sets in Han et la. | Acetylation* | 49 (55) | 867 (868) |
| | Glycosylation | 41 | 711 |
| | Hydroxylation | 26 (27) | 133 |
| | Methylation | 23 (27) | 171 |
| | Phosphorylation* | 68 (79) | 389 |
| IEDB data sets | | 5,711 (5,712) | 15,196 (15,198) |

### Table 2 - Classification performance on the PTM data sets

A classifier WB-SVM-IG by Han et al. is an SVM classifier using words selected for IG > 0.02 as features. SB-NB-WRST is a Naïve Bayes classifier using substrings of words selected for Wilcoxon rank-sum test ≥ 0.15 as features. For the results we obtained, we report AUC and BEP measures for the best performing single SVM and SVM ensemble classifiers with the settings of R, a percentage of words used as features (see the Method section).

| | | Acetylation | | Phosphorylation | |
|---|---|---|---|---|---|
| **Methods** | | **AUC** | **BEP** | **AUC** | **BEP** |
| WB-SVM-IG by Han et al. | | .869 | n/a | .896 | n/a |
| SB-NB-WRST by Han et al. | | **.916** | n/a | .925 | n/a |
| Single | (R%) | .892 (4) | .440 (9) | .923 (4) | .643 (9) |
| Ensemble | (from 1 up to R%) | .913 (25) | **.509** (25) | **.931** (16) | **.677** (16) |

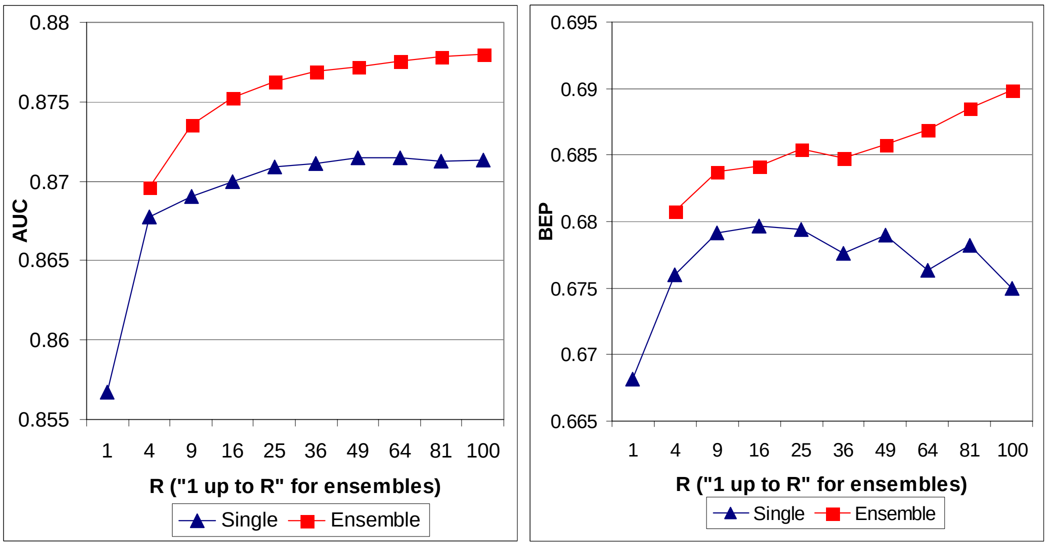
### Table 3 - Classification performance on the IEDB data sets

The classifier (NB) by Wang et al. employed a Naïve Bayes method using around 20,000 word features selected for document frequency (DF) >3 and IG >0.00002. Another classifier by Wang et al. (NB w/ MeSH etc.) uses additional features such as MeSH headings and other domain-oriented/task-specific features. See the caption of Table 2 for the definition of R used in the classifiers we trained.

| **Methods** | | **AUC** | **BEP** |
|---|---|---|---|
| NB by Wang et al. | | .838 | n/a |
| NB w/ MeSH etc. by Wang et al. | | .848 | n/a |
| NB | (R%) | .837 (49) | .645 (25) |
| NB ensemble | (from 1 up to R%) | .840 (100) | .639 (64) |
| SVM | (R%) | .871 (49) | .680 (16) |
| SVM ensemble | (from 1 up to R%) | **.878** (100) | **.690** (100) |

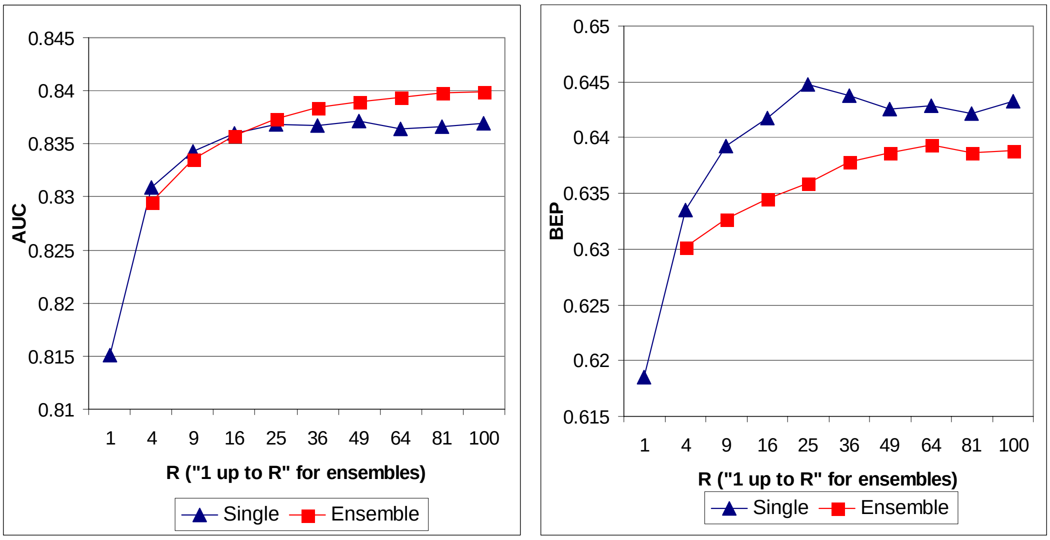
## Figure 1 - Classification performance on the IEDB data sets (SVM)

Each figure shows how AUC or BEP changes as the setting of R changes for single SVM classifiers (▲) and SVM ensemble classifiers (■).



## Figure 2 - Classification performance on the IEDB data sets (Naïve Bayes)

The each figure shows how AUC or BEP changes as the setting of R changes for single Naïve Bayes classifiers (▲) and Naïve Bayes ensemble classifiers (■).


AUC
R ("1 up to R" for ensembles)
Single
Ensemble
BEP
R ("1 up to R" for ensembles)
Single
Ensemble