Hybrid Feature Selection Methods for Online Biomedical Publication Classification

CONFERENCE PAPER · AUGUST 2015

7 AUTHORS, INCLUDING:

Yanqing Zhang
Georgia State University
18 PUBLICATIONS **151** CITATIONS

SEE PROFILE

Angela R Laird
Florida International University
154 PUBLICATIONS **11,999** CITATIONS

SEE PROFILE

Jessica Ann Turner
Georgia State University
171 PUBLICATIONS **3,669** CITATIONS

SEE PROFILE

Matthew Turner
Georgia State University
12 PUBLICATIONS **22** CITATIONS

SEE PROFILE

Hybrid Feature Selection Methods for Online Biomedical Publication Classification

Long Ma, Yanqing Zhang,
Raj Sunderraman
Department of Computer Science
Georgia State University
Atlanta, Georgia, USA
lma5@student.gsu.edu
yzhang@gsu.edu
raj@cs.gsu.edu

Peter T. Fox
University of Texas Health Science Center
San Antonio (UTHSCSA), USA
fox@uthscsa.edu

Angela R. Laird
Department of Physics
Florida International University
Miami, Florida, USA
angie.laird@fiu.edu

Jessica A. Turner, Matthew D. Turner
Departments of Psychology & Computer Science
Georgia State University
Atlanta, Georgia, USA
jturner@mrn.org
mturner46@gsu.edu

Abstract—We review several feature selection methods: Recursive Feature Elimination, Select K Best, and Random Forests, as elements of a processing chain for feature selection in a text mining task. The text mining task is a multi-label classification problem of label assignment; metadata that is usually applied to published scientific papers by expert curators. In the formulation of this classification task, a feature space that is dramatically larger than the available training data occurs naturally and inevitably. We explore ways to reduce the dimension of the feature space, and show that sequential feature selection does substantially improve performance for this complex type of data.

Keywords: multi-label classification, text mining, feature selection, metadata annotation, hybrid feature selection

I. INTRODUCTION

In order to properly classify the published biomedical sciences research literature, it is essential to have automated systems that can correctly label the technical aspects of the published papers with appropriate metadata. This allows researchers to ask precise technical questions of this literature. Labels are essential as the use of simple keyword search does not sufficiently identify relevant papers [1]. Currently such labeling requires the time and attention of human scientific experts who are usually neither interested in carrying out the labeling task nor are they freely available to do it. As an example, we work with experts who label published scientific papers with metadata that identifies key aspects of the experimental designs, subject populations, and methods (among other things) used in the papers. This is part of their ongoing research that involves the construction of large-scale meta-analytic reviews of the neuroscience literature. While they are interested in doing the labeling task for papers they use directly as input to their work, they are scientists, not curators, and have little direct interest in labeling papers they do not use. As part of their larger overall workload, it can take months to label just a hundred papers. Additionally, their work is really conducting the meta-analysis and literature review; in a more perfect scientific literature, the metadata would already be part of the publications. Our research program addresses this need, but using machine learning approaches to automate the process of assigning metadata based on the raw text of the scientific literature.

The specific problem addressed in this work is how to preprocess the scientific vocabulary used in publications into a useful feature space for machine learners. The problem is two-fold. First: there is no pre-existing, agreed-upon, and fixed vocabulary in use in the sciences that guarantees that important ideas are always expressed in identical terms. Second: the deeper ideas in research are not naturally expressed in simple "keyword" terms; they are concepts that require multi-word explanations that contain and combine multiple underlying concepts. Our problem is to mine the scientific texts for the words that indicate these underlying concepts and then assign metadata labels that make these concepts explicit. Simple searching for

This project was supported by grants from NIMH/NIH: R56-MH097870, R01-MH084812, and R01-MH074457.
978-1-4799-6926-5/15/$31.00 ©2015 IEEE

a fixed list of keywords will not suffice for this complex text structure. The metadata must create syntax to mark the specific semantic content in the texts.

In this situation, it is tempting to remove only the usual stop words (short function words which express grammatical structure; e.g., propositions, pronouns, articles, and particles) for the language in question. However, this leaves a collection of features that is too large to use effectively as a starting point for applying standard text mining algorithms as many algorithms work better with smaller feature spaces [2]. This problem is particularly acute when the number of training examples is very small, as often happens in the case of expertly labeled documents used for supervised learning classification problems. This is an example of an $n \ll p$ (n much less than p) problem: any problem where the collection of training instances (n) is much less than the number of features or parameters (p) [3, 4].

In particular, many types of high-quality data suffer from this $n \ll p$ phenomenon, while most methods in data mining were invented for situations where there is an abundance of relatively crude data. In our work, we have 3,606 unique words after stop word removal, but the number of training instances to be used in the text classification task is only 247. So our data has more than an order of magnitude more features than training instances (a factor of approximately 15).

Our classification problem is inherently a “multi-label” problem [5, 6]. This is a type of problem where each instance can be labeled with any possible combination of the labels available for the task. In recent work [7], we approached this problem by using a multi-instance multi-label (MIML) algorithm [8]. There we used principal component analysis (PCA) and vocabulary stemming to pre-process the data before classification [9-11]. It was shown that the PCA components, used as features, allowed a dramatic reduction in feature space dimension. However, as reported there, that may have been an artifact of the particular data used in that study. To address that issue, we use here a broader collection of data.

We try the approach of reducing the original feature space directly. We consider a variety of feature selection and feature reduction strategies, and use each of these in various combinations, to prepare the data for classification. We use an off-the-shelf text classification method, a support vector machine (SVM) [12, 13]. To allow this binary classifier to be used for a multi-label problem, we use a *problem transformation* method, called *binary relevance*, to convert this problem to the multi-label context [5]. The main novelty in this approach is in combining the feature selection methods sequentially to improve their performance.

II. METHODS

A. Data Preparation and Preprocessing

The training data are the text of the published abstracts of 247 human neuroimaging journal articles and their corresponding metadata labels. These metadata labels were created as part of the BrainMap database (www.brainmap.org) and were annotated using the standard set of Cognitive Paradigm labels (www.cogpo.org), identifying aspects of the experimental designs reported in each of the papers [6, 14-16]. These labels were originally added to the BrainMap database by trained expert annotators, based on their reading of the entire text of each journal article. The classifiers developed here only use the partial information provided by the article abstract text, not the full article text; therefore we expect some crucial detail to be lost and this will be reflected in lower performance scores.

The specific metadata we are using has been assigned to articles in previous work, primarily by one of the present authors (ARL). This careful expert assignment is used as a gold standard for training the classifier. The ultimate goal of this research is to develop fully automatic classifiers that can perform at levels competitive with human experts. This is essential as the classification task is both (1) dependent on significant expertise while also (2) being repetitive and boring; the former issue limiting the number of potential curators able to do the task, and the latter leading to potential errors and overall work dissatisfaction for the curators.

In our experiments, there are eight label dimensions (label sets) and each dimension has multiple labels represented in our data: Paradigm Class Labels (PCL, 48 labels), Behavior Domain Labels (BDL, 40 labels), Stimulus Type Labels (STL, 17 labels), Instruction Type Labels (ITL, 14 labels), Disease Labels (DL, 13 labels), Response Type Labels (RTL, 9 labels), Stimulus Modality Labels (SML, 5 labels), and Response Modality Labels (RML, 5 labels). As this is a multi-label problem, each training instance may have more than one label. For instance, in the ITL dimension, which describes details of how research subjects receive instructions in cognitive neuroscience experiments, the average number of labels assigned across the 247 instances is 1.65 (mostly 1 or 2 labels per instance) and with a maximum of 6 labels assigned to any instance. Fuller details for most label dimensions can be found in [16] and a more complete description of this data set is available in [6].

First the stop words and punctuation were removed from the texts and the words tokenized using the Natural Language Tool Kit (NLTK; www.nltk.org) [9, 17]. Each abstract is ultimately reduced to a count-based bag-of-words vector representation.

After the punctuation and stop word removal and tokenization the base vocabulary is 3,606 words. Before forming the word counts, we further reduce the number of words by using word stemming as a preprocessing step. This procedure maps morphological variants onto their common stems. We used NLTK for word stemming and lemmatizing. We used the Porter Stemmer and the WordNet Lemmatizer, both of which are parts of the NLTK Stem package. The Porter stemmer is not very aggressive and will leave many variants unchanged [10]. Additionally, we removed any words strictly less than length 3. The stems, along with any words left untouched by the stemming and other processes, constitute our vocabulary; i.e. these are the words for the word counting process. This combination of procedures reduced the feature dimension from 3,606 to 2,317 words.

This results in a final data matrix with 2,317 rows (one per word) and 247 columns (one per abstract; these are the count vectors) and with the counts of words present in each abstract populating the body of the matrix. We use this as the input for the processes below. Note that most abstracts are lengths from 100-300 words, so these vectors are sparse; each vector will have less than 10% non-zero elements.

B. Feature Selection Methods

We use the following three feature selection methods as our basis:

(1) Recursive Feature Elimination (RFE; also called RFE-SVM), recursively prunes features according to each feature's importance [18, 19]. Feature importance is defined here as the weight given to the feature by an SVM classifier. Feature importance is determined by sequentially re-training a SVM classifier and, at each step, removing less useful features, i.e. features below some threshold weight. The RFE process continues until the target number of features remains.

(2) Select K Best (SKB) is a procedure that constructs the χ^2 (chi-square) statistic between each element of the feature space and each of the labels to determine which features are correlated with which labels [20]. Then it removes the least significant features, which are less likely to be useful in any classification task. Note that it rejects features with the smallest χ^2 statistics, which is different from, but similar to, rejecting based on Pearson- or Spearman-type correlations; the χ^2 is more convenient as a one-sided, strictly positive measure [21, 22].

(3) Random Forests (RF) select features at random (with replacement) and group each subset in a random subspace [23-25]. It is known that some ensemble learning methods for classification or regression can be used as feature ranking methods if a relevant importance score can be defined. Here we set the number of trees to 10 and record the percentage of trees in which a given features appears; this we use as the importance score. The averages of these scores for each feature order the features by importance, and this allows ranking of features, making elimination of less important features trivial.

In previous work, various ways of combining feature selection techniques have been proposed. For instance, Li et al. [26, 27] show that a combination of feature scoring and ranking methods can outperform individual feature scoring when appropriate adjustments are made to the scores arising from the different methods (e.g. normalization, conversion to ranks). In [28], Neumayer et al. show the results of various combinations of feature selection methods. The individual methods they have used include document frequency, information gain, GSS-Coefficient (a variant on the χ^2 statistic, see [29]), among others. While [28] does not show a distinct improvement in performance over individual feature selection methods, [30] does. In this latter study, simple strategies based on mathematically combining ranking scores (the χ^2 in most cases) creates better feature selection evaluation over individual scores. There is, therefore, evidence of improvement with combined methods, but simultaneously no clear winning method across all data sets. Based on this work, we propose to explore whether combinations of our feature selection methods can improve performance over the use of single methods.

Our primary interest is in the RFE and SKB methods, as these are the most heavily used methods in prior work. We tried using them both singly, in sequence, and in both possible orders. The first method in a sequence selects a subset of the original 2,317 words, and then the second selects a smaller subset using the output of the first algorithm as input. We then used RF as a classifier to select feature subsets with the procedure described above. Because RF generates random subspaces, it was not used as the first method for feature selection [25]. This is due primarily to the randomization (with replacement) and missing features from the original feature space. However, RF can be used efficiently after either of the other methods, or their combination.

C. Multi-Label Classification and LinearSVC

One approach to the multi-label classification problem is to convert a multi-label classification task into a set of single-label or binary classification tasks, and there are a variety of ways to do this [5, 31, 32].

One method, binary relevance, decomposes multi-label problems with *n* labels into *n* independent binary classification tasks [5, 6, 33]. In this method labels are predicted independently and so the label dependencies are ignored completely [34]. However, ignoring dependence is usually not a substantial detriment to performance, whether it is done to the features or the labels, in general text classification [35, 36]. In order to use binary relevance method, we implement One-versus-the-Rest (OvR) multi-label strategy on our training data [21]. In this method, an individual classifier is built for each label to be predicted, and the entire data set is partitioned into cases with the label and cases without the label. The partition is done without respect to any of the other labels.

Binary relevance is a meta-method; a procedure for decomposing a multi-label problem into a set of binary problems. As such we still need to specify a classifier to use with the decomposition to complete the classification task. We use the LinearSVC (linear support vector classifier) implementation of the linear support vector machine in Scikit-Learn [17]. The classifier is used with settings determined in previous research with the same data [6]. This standard classifier is based on the LIBLINEAR implementation [37].

III. RESULTS

In our performance analysis, we considered 8 possible methods and combinations of methods. Our target goals were to reduce the 2,317 starting features by at least an order of magnitude, so we considered reduced feature sets of sizes: 50, 100, 150, 200, and 250; in some cases these values are the actual final feature set sizes, in other cases these are nominal (maximum possible) sizes, see below for more details.

A. Feature Selection Methods and Chains of Methods

In this section, we implemented three feature selection methods on our training data: RFE, SKB and RF. We evaluated the RFE and SKB individually and combined them as sequential approaches, as well as using each of them as a preprocessing step for RF.

The methods and combinations are as follows:

(1) **RFE** – The RFE method is used as the only feature selection method. It was run five times, selecting either 50, 100, 150, 200, 250 features and these were used as input for the BR Linear SVC classifier.
(2) **SKB** – The SKB method is used as the only feature selection method. The details are as for the **RFE** method.
(3) **RFE(300)→SKB** – First, the RFE method selects the 300 most important features; then from these 300 features, SKB selected the 50, 100, 150, 200, 250 most important of these. These five final feature sets are used for the classification as above.
(4) **SKB(300)→RFE** – The same as the previous procedure, but in the other order.
(5) **RFE(50-250)→RF** – In this condition, RFE selects a fixed size subset of the original features; then RF is applied to this subset. Because of the random nature of RF, the final sizes of the feature subsets it selects are variable; the nominal sizes (produced by the RFE step) are reported in Fig. 1. The actual final feature sets were all much smaller, ranging from 14 to 73 features after the RF step; varying by label dimension (Fig. 2).
(6) **SKB(50-250)→RF** – This is the same as the previous procedure, except with SKB as the first feature selector. Here the actual number of features produced by the RF step ranged from 7 to 68; these also varied by dimension.

Fig. 1 shows classification performance on the eight label dimensions (see II.A. for the identifiers of label sets) using the above six feature selection methods and combinations.

The results presented are 10-fold cross-validated F1-Micro scores, see [6, 7] for details of the cross-validation procedures and [6] for more on F1-Micro as a performance measure. Binary measures of performance are not sufficient for multi-label classification *per se*; they can be applied to each label individually, but cannot be used for overall performance. We use the F1-Micro score which is a balanced combination of precision (also called positive predictive value; the proportion of relevant retrieved labels to the total set of retrieved labels) and recall (sensitivity; the proportion of retrieved relevant labels relative to relevant labels available to be retreived). F1 varies between 0 and 1, with scores closer to 1 being better. Micro averaging the F1 score, across test or training instances, allows for better comparisons across data sets; however more complex data sets will intrinsically have lower F1 scores.

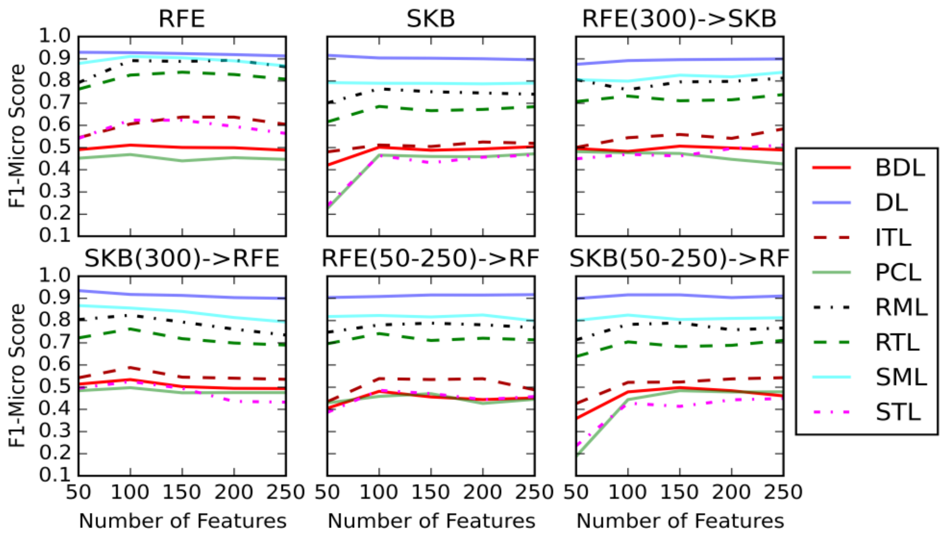

Fig. 1. Feature Selection Methods on Eight Label Dimensions

The results in Fig. 1 show there is no profound difference among the feature selection methods when applied to our training data. Both **RFE** and **SKB(300)→RFE** do better overall than the other methods, but most methods show some improvement over the same classification done with the entire original feature space as input. (See Table I and the discussion of overall results in section III.C below for more on best performance and performance using all of the features, without any feature selection being applied.)

The last two conditions, **SKB(50-250)→RF** and **RFE(50-250)→RF** are shown in Fig. 1 in terms of nominal (input) numbers of features. For comparison, Fig. 2 shows the same results plotted in terms of actual numbers of features. The main thing to note here is that the actual number of features in use in either of these conditions is always less than about 75; see the descriptions of the methods above for maximum and minimum values.

B. Common Features Selected by Multiple Methods

Because each feature selection method chooses features by different statistical tests, it is reasonable to expect that the selected features might differ. However, features chosen by multiple feature selection methods might also be more reliable. We considered two different combinations of this type:

(7) **Common Features (RFE&SKB)** – we first selected 50, 100, 150, 200, 250 features using **RFE** and corresponding numbers from **SKB** and then paired up the corresponding sets. We obtained the intersection of the sets, and the features falling into the intersection were used as features for training the LinearSVC classifier.

(8) **Common Features (RFE&SKB&RF)** – Same as the previous method, but also with the top ranked 50, 100, 150, 200, 250 features from RF and a three-way set intersection.

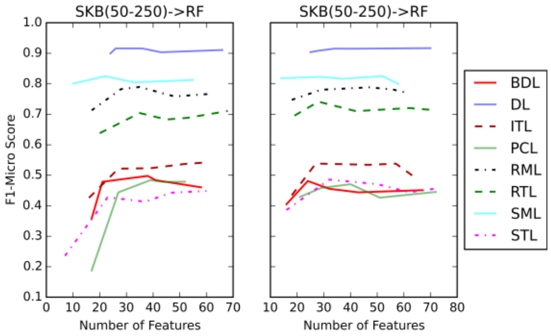

Fig.2. Feature Selection Methods Using RF

In Fig. 3, the *x*-axis is the number of features (resulting from the set intersections) and *y*-axis is the F1-Micro score obtained by the classifier using the co-occurrence feature sets. The left-hand side shows the co-occurrence features in five feature sets (50-250) that are selected by **RFE&SKB**; while, the right hand side shows the co-occurrence features from the three feature selection methods, **RFE&SKB&RF**. In general, the performance improves as the number of number of common features increases. This figure is plotted in terms of the actual number of features used for the classifier, not in the nominal 50-250 range. It is important to note the number of features in these conditions is much smaller than most other conditions presented here; in the left panel the maximum number of common features is 117, on the right it is 51.

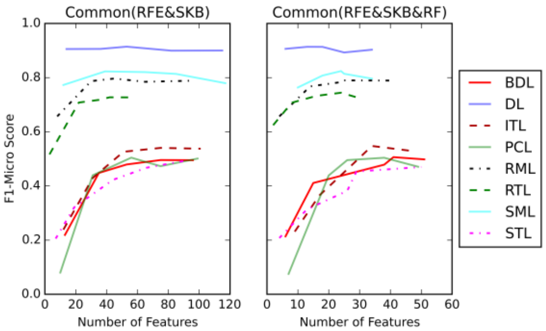

Fig.3. Co-occurrence features among eight label dimensions.

Performance at the largest number of features in Fig. 3 is in the general range of performance in the previous figures, but there is no obvious improvement over the previous results. On only one label dimension (PCL or Paradigm Class Labels) did this collection of methods achieve better performance than that seen in other conditions. See Table I below. It is worth noting that PCL is a conceptually complex dimension compared to most of the others.

C. Overall Performance

Table I summarizes the overall results of the study, including a major point that may not be clear from the presentation above: feature selection does improve performance overall. In Table I, the first row shows the F1-Micro score obtained by the classifier when applied to the entire feature space of 2,317 words ("All Features") without any feature selection. The second row shows the best F1-Micro score obtained across all methods applied above. The third row shows the improvement in F1 score due to feature selection. In every case, at least one of the feature selection strategies tried above was able to provide a smaller feature set to the classifier that also improved the classifier's performance; and often it is a dramatic improvement. The smallest improvement is 0.0519 F1-Micro units (highlighted) in the disease label (DL) dimension. This dimension is generally "easier" for classifiers to do well in solving, as the disease labels themselves are often explicit features present in the input text. The best improvement is in the stimulus type (STL) dimension, with an improvement of 0.2251 F1-Micro units. Note also that the behavioral domain labels (BDL) and paradigm class labels (PCL) both show between 0.10 and 0.14 F1-Micro units of improvement. Conceptually, these are the hardest dimensions, depending on sometimes complex combinations of features in the input text to represent these labels' presence or absence. See [6] for more details. The last two rows give the winning method and the number of features (actual) needed to achieve the score. In general, **RFE** alone and **RFE** preceded by **SKB** achieved the best results. No more than 200 features were used.

Another point to consider in the table: if we set an F1-Micro score of 0.50 as a conservative cut-off for the usefulness of a classifier (see [6] for a discussion of this point), we find that using the original feature space we only have useful classifier results in 4 out of 8, or half, of our cases. With feature reduction, whether it is hybrid or just single methods, all 8 of the label dimensions now have useful classifier results, with 6 of the 8 being well above this mark.

IV. CONCLUSIONS

This paper presents an approach to substantial feature reduction in large complex textual feature spaces using standard methods both alone and in combination. The overall result shows that one standard method, RFE, used alone often is the best choice for feature reduction (here it has the best performance in 5 of 8 cases). However, combining methods produces a better result in three cases, including two label dimensions with the most complex conceptual structure (BDL and PCL). Additionally, a combined method produced the maximal improvement in one dimension (DL) where there was very little room for improvement. These novel results encourage the continuing exploration of hybrid feature reduction methods in the future for other data sets.

Our research program focuses on the $n \ll p$ problem, that is, the problem of classifying very small quantities of very high-quality, high-dimensional data. In this case, we have very expensive to produce data that can only come from intense expert work. This type of data will always have an imbalance between n and p, that is, there will always be many more features than training instances.

Performance of the classification algorithm on the raw feature space is not very good, but with several combinations of feature selection methods, the performance was improved substantially. We are currently developing these techniques using both more data (a larger sample of the BrainMap database), different types of labels with different underlying structure and complexity, and using the full-text of the scientific articles rather than just the text of the abstracts. It should be noted that this last goal, the using of full text, will actually make the $n \ll p$ problem worse: even with a larger corpus of expert assigned labels, the number of features will dramatically increase if we use the full text of the scientific papers. With this sort of data, this imbalance is a permanent feature of the data.

TABLE I. F1-MICRO SCORES WITH WINNING METHODS

	Label Dimensions							
	BDL	***DL***	***ITL***	***PCL***	***RML***	***RTL***	***SML***	***STL***
F1-Micro Score (All Features)	0.4294	0.8830	0.4746	0.3672	0.7187	0.6808	0.7418	0.3983
Best F1-Micro Score	0.5340	0.9349	0.6374	0.5044	0.8936	0.8395	0.9107	0.6234
Difference (max – min)	0.1046	**0.0519**	0.1628	0.1372	0.1749	0.1587	0.1689	**0.2251**
Method	SKB(300) →RFE	SKB(300) →RFE	RFE	Common (RFE&SKB)	RFE	RFE	RFE	RFE
Number of Selected Features	100	50	150	56	200	150	100	100

Currently, text annotation takes a great deal of time and effort by the limited number of humans available with the requisite knowledge to do the task. While the production of scientific knowledge has continued to increase, the availability of tools to manage this knowledge is still not sufficiently developed. This is a substantial problem with connections both to deeper issues in machine learning and also to interesting technical problems.

ACKNOWLEDGMENT

This project was made possible by a collaboration agreement allowing comprehensive access to the BrainMap database, a copyrighted electronic compilation owned by the University of Texas.

REFERENCES

[1] T. Gross and A. G. Taylor, "What have we got to lose? The effect of controlled vocabulary on keyword searching results," *College & Research Libraries,* vol. 66, pp. 212-230, 2005.

[2] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*: Morgan Kaufmann, 2005.

[3] E. Candes and T. Tao, "The Dantzig selector: Statistical estimation when p is much larger than n," *Annals of Statistics,* vol. 35, pp. 2313-2351, Dec 2007.

[4] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, "The elements of statistical learning: data mining, inference and prediction," *The Mathematical Intelligencer,* vol. 27, pp. 83-85, 2005.

[5] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *Dept. of Informatics, Aristotle University of Thessaloniki, Greece,* 2006.

[6] M. D. Turner, C. Chakrabarti, T. B. Jones, J. F. Xu, P. T. Fox, G. F. Luger, *et al.*, "Automated annotation of functional imaging experiments via multi-label classification," *Frontiers in neuroscience,* vol. 7, 2013.

[7] D. Ren, L. Ma, Y. Zhang, R. Sunderraman, P. T. Fox, A. R. Laird, *et al.*, "Online biomedical publication classification using multi-instance multi-label algorithms with feature reduction," in *14th IEEE International Conference on Cognitive Informatics and Cognitive Computing (ICCI*CC2015)*, Beijing, China, 2015.

[8] S.-J. Huang and Z.-H. Zhou, "Fast Multi-Instance Multi-Label Learning," *arXiv: 1310.2049,* 2013.

[9] *Natural Language Tool Kit (NLTK) 3.0 documentation*. www.nltk.org/api/nltk.stem.html

[10] M. F. Porter, "An algorithm for suffix stripping," *Program,* vol. 14, pp. 130-137, 1980.

[11] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems,* vol. 2, pp. 37-52, 1987.

[12] J. A. K. Suyken and J. Vandewalle, "Least Squares Support Vector Machine Classifiers," *Neural Processing Letters,* vol. 9, pp. 293-300, 1999.

[13] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *J. Mach. Learn. Res.,* vol. 2, pp. 45-66, 2002.

[14] P. T. Fox, A. R. Laird, S. P. Fox, P. M. Fox, A. M. Uecker, M. Crank, *et al.*, "BrainMap taxonomy of experimental design: description and evaluation," *Hum Brain Mapp,* vol. 25, pp. 185-98, May 2005.

[15] P. T. Fox and J. L. Lancaster, "Opinion: Mapping context and content: the BrainMap model," *Nat Rev Neurosci,* vol. 3, pp. 319-21, Apr 2002.

[16] J. A. Turner and A. R. Laird, "The cognitive paradigm ontology: design and application," *Neuroinformatics,* vol. 10, pp. 57-66, 2012.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research,* vol. 12, pp. 2825-2830, Oct 2011.

[18] J. Bedo, C. Sanderson, and A. Kowalczyk, "An Efficient Alternative to SVM Based Recursive Feature Elimination with Applications in Natural Language Processing and Bioinformatics," in *AI 2006: Advances in Artificial Intelligence*. vol. 4304, A. Sattar and B.-h. Kang, Eds., ed: Springer Berlin Heidelberg, 2006, pp. 170-180.

[19] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine learning,* vol. 46, pp. 389-422, 2002.

[20] A. Abraham, F. Pedregosa, M. Eickenberg, P. Gervais, A. Mueller, J. Kossaifi*, et al.*, "Machine learning for neuroimaging with scikit-learn," *Front Neuroinform,* vol. 8, p. 14, 2014.

[21] *Scikit-learn*. http://scikit-learn.org/stable/

[22] K. A. Markus, "Principles and Practice of Structural Equation Modeling, 3rd edition," *Structural Equation Modeling-a Multidisciplinary Journal,* vol. 19, pp. 509-512, 2012.

[23] L. Breiman, "Random forests," *Machine learning,* vol. 45, pp. 5-32, 2001.

[24] R. Genuer, J. M. Poggi, and C. Tuleau-Malot, "Variable selection using random forests," *Pattern Recognition Letters,* vol. 31, pp. 2225-2236, 2010.

[25] A. Liaw and M. Wiener, "Classification and Regression by randomForest," *R News,* vol. 2, pp. 18-22, 2002.

[26] Y. Li, D. F. Hsu, and S. M. Chung, "Combining multiple feature selection methods for text categorization by using rank-score characteristics," in *21st International Conference on Tools with Artificial Intelligence, 2009. ICTAI'09.* , 2009, pp. 508-517.

[27] Y. Li, D. F. Hsu, and S. m. Chung, "Combination of multiple feature selection methods for text categorization by using combinatorial fusion analysis and rank-score characteristic," *International Journal on Artificial Intelligence Tools,* vol. 22, p. 1350001, 2013.

[28] R. Neumayer, R. Mayer, and K. Nørvåg, "Combination of feature selection methods for text categorisation," in *Advances in Information Retrieval*, ed: Springer, 2011, pp. 763-766.

[29] Y. Liu, "A Comparative Study on Feature Selection Methods for Drug Discovery," *Journal of Chemical Information and Computer Sciences,* vol. 44, pp. 1823-1828, 2004.

[30] J. O. S. Olsson and D. W. Oard, "Combining feature selectors for text classification," presented at the Proceedings of the 15th ACM international conference on Information and knowledge management, Arlington, Virginia, USA, 2006.

[31] J. Fürnkranz, E. Hüllermeier, E. L. Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *Machine learning,* vol. 73, pp. 133-153, 2008.

[32] S. Godbole and S. Sarawagi, "Discriminative methods for multi-labeled classification," in *Advances in Knowledge Discovery and Data Mining*, ed: Springer, 2004, pp. 22-30.

[33] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine learning,* vol. 85, pp. 333-359, 2011.

[34] E. Montañes, R. Senge, J. Barranquero, J. R. Quevedo, J. J. del Coz, and E. Hüllermeier, "Dependent binary relevance models for multi-label classification," *Pattern Recognition,* vol. 47, pp. 1494-1508, 2014.

[35] H. Zhang, "The optimality of naive Bayes," *AAAI,* vol. 1, p. 3, 2004.

[36] H. Zhang, "Exploring Conditions For The Optimality Of Naïve Bayes," *International Journal of Pattern Recognition and Artificial Intelligence,* vol. 19, pp. 183-198, 2005.

[37] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin, "LIBLINEAR: A Library for Large Linear Classification," *Journal of Machine Learning Research,* vol. 9, pp. 1871-1874, Aug 2008.