Data duplication: an imbalance problem ?

Aleksander Kołcz A.KOLCZ@IEEE.ORG
Abdur Chowdhury CABDUR@AOL.COM
Joshua Alspector JALSPECTOR1@AOL.COM
AOL, Inc., 44900 Prentice Drive, Dulles, VA 20166 USA

Abstract

The task of finding duplicate records in large databases has long been an area of active research in the data-mining community. The gravity of the problem posed by duplicate "contamination" of the data has rarely been addressed directly, however, with more efforts being directed at the related problem of class imbalance. We consider the consequences of duplicate presence on the quality of classifiers learnt with such data, emphasizing that contamination rates among the training and test collections may be considerably different. The discussion is supported by experiments using the spam-detection task as an example where dealing with varying degrees of duplication is commonplace. Our results indicate the generally detrimental effect of duplicate presence on classification accuracy, but also demonstrate that, for classifiers such as Naive Bayes Perceptron with Margins, duplicate rates need to be rather high to be truly harmful.

1. Introduction

Many practical problems in predictive data mining utilize data acquired from different sources and/or over different periods of time. It has long been recognized that, aside from other quality control issues, such process can lead to accounting for essentially the same record multiple times. To counter this phenomenon (referred to, among others, as record linkage [40], reference matching [26], deduplication [36] or copy detection [37]), a number of solutions have been proposed, ranging from manually coded rules to applications of the latest machine learning techniques. Their accuracy varies and, for large collections, some of these techniques may be computationally too expensive to be deployed in their full capacity. In reality, despite best efforts to clean the data, duplicates may be present and it is important to understand their impact on the quality of the data mining performed. I particular, the presence of duplicates may skew the content distribution of the learning sample and thus affect the process of classification in a way similar to that observed in class-imbalance scenarios [18]. Unlike the impact of general class imbalance, the practical consequces of data duplication appear to be less understood however.

In this work we investigate the effects of data duplication on classifier accuracy using the problem of spam detection as a real-world example (although the results should be applicable to other domains). Spam filtering is viewed as a two-class text categorization problem, where data duplication occurs naturally due to messages being sent in volume (particularly for marketing and commercial purposes). Due to the temporal nature of email, the distributions of duplicates in the learning sample and the target environment will generally differ[1]. Also, measures of duplicate detection are actively countered (by spammers), which reduces the effectiveness of content-based deduplication [15]. In this report we identify the potentially harmful effects of the duplicates on the process of classifier learning and evaluation and provide experimental results based on actual email data.

The paper is organized as follows: Section 2 discusses the sources of duplicates in email data. In Section 3 we outline the potential effects of duplicate presence on classifier learning, while an evaluation scheme based upon unique data is presented in Section 4. Section

[1] We consider the problem of spam detection from the system-wide perspective, where a single filter needs to serve a large group of people. The related problem of spam detection on a personal level is not significantly affected by data duplication and is not discussed in this work.

Workshop on Learning from Imbalanced Datasets II, ICML, Washington DC, 2003.

5 describes an experimental setup for evaluating the impact of duplication rates on classification accuracy. The results are presented in Section 6. Section 7 describes prior work related to learning with imbalanced data, spam filtering and detection of duplicate text documents. The paper is concluded in Section 8.

2. Duplicates, volume and the sample selection bias

Traditionally, it is assumed that the collection of data available to a learner represents an i.i.d. sample from the underlaying probability distribution. This is often violated in practice, and the learning sample may exhibit properties quite different from the ones encountered by the classifier once deployed [42]. Often this is caused by the sample selection bias, whereby certain areas of the input space are more costly to sample than others. For example, in the email classification domain obtaining labeled instances of personal/sensitive emails is very difficult due to understandable privacy concerns but, ironically, private emails are also considered to be the most valuable to the users, and the classifier faces the difficult problem of insuring a low error rate in an area of the input space that is extremely undersampled. On the other hand, it is easy to establish a data collection process where primarily spam and legitimate *bulk* messages are collected. Here, essentially the same message may be encountered multiple times, where the duplication rate in the collected sample does not necessarily have to reflect the original duplication rate in the email stream (unless the data collection process insures uniform sampling), and certainly not the future expected duplication rate.

For on-line learning scenarios with fair sampling one could argue that the presence of duplicates represents useful information about the target invironment, which should be taken into account when adjusting the model. We consider however the more typical scenario where the learning sample is collected over an extended period of time, in a setup possibly quite different from the target environment of the classifier. Thus if indeed duplicates in the test sample are present, they are not necessarily the same (or even of similar content) as the ones present in the learning sample.

3. Classifier learning in the presence of duplicates

3.1. Cost-sensitive classification

According to the Bayesian decision theory, the quality of any particular classifier F in an N-class discrete-input setting (i.e., $F : \mathcal{X} \rightarrow \{1..N\}$) can be expressed by the loss function [12]

$$\mathfrak{L} = \sum_{x} P(x) \sum_{i=1}^{N} [F(x) = i] \sum_{j=1}^{N} P(j|x)\, C(j, i, x) \quad (1)$$

where, $P(j|x)$ is the probability that x belongs to class j and $C(i, j, x)$ is the cost associated with classifying x as a member of class i when in fact it belongs to class j; $F(x)$ denotes the decision made by the classifier, with $[F(x) = i]$ equal to 1 if the classifier returns i as the class label – and is otherwise equal to 0. Usually, it is assumed that predicting the correct class leads to zero cost.

For the particular 2-class problem of spam filtering we denote the class labels as `L` (`legit`) and `S` (`spam`), and (1) is transformed to

$$\mathfrak{L} = \sum_{x} P(x) \left(P(\mathtt{S}|x) [F(x) = \mathtt{L}] + \mathit{cost}(x) \cdot P(\mathtt{L}|x) [F(x) = \mathtt{S}] \right) \quad (2)$$

where *cost* (x) is the cost of misclassifying a legitimate message as spam, while it is assumed the cost of misclassifying a spam message as legitimate to be unity. This is because the cost of letting the spam through is fairly nominal (and linked to the cost of physical storage and the resources being spent on handling customer complaints).

The cost of a "civilian kill" is usually considered to be higher, since the loss of an important email might mean more than mere annoyance to a customer. The precise value of *cost* (x) is difficult to measure however, and the literature provides many examples where it is simply set to a certain "reasonable" number (e.g., 10–100) [32][1]. A possible "compromise" is to forgo the concept of message-specific misclassification costs and only to differentiate the cost of misclassifying specific sub-categories of legitimate email (e.g., personal vs. bulk)[19]. This will not be considered here however and, for simplicity, we will assume that *cost* (x) is message independent, i.e., $\mathit{cost}(x) = \mathit{cost}$ for all x.

3.2. Potential impact of duplicates

3.2.1. Classifier learning

If the learning collection of data is used to optimize F (and it's particular operating point, if applicable) such that $\mathfrak{L}$ is minimized, it is clear that duplicates will be influential insofar as they affect the estimates of $P(x)$, $P(\mathtt{S}|x)$ and $P(\mathtt{L}|x)$, and the actual process of learning F, which may be algorithm dependent. Note that in many cases the optimization process will induce F first, using the learning sample, and then (2) gets applied to adjust the final decision threshold. High rates of duplication are likely to increase the importance of misclassification costs in the corresponding regions of the input space $\mathcal{X}$ (due to relatively high values of $P(x)$ estimated), and since many learners are naturally minimizing the error rate, they might overfit the corresponding areas of $\mathcal{X}$. Note that in direct cost-sensitive decision making [42], the classification outcome depends only on the values of $P(\mathtt{S}|x)$ and $P(\mathtt{L}|x)$ and *cost*. In this case, duplicates impact classification via their influence on estimation of $P(\mathtt{S}|x)$ and $P(\mathtt{L}|x)$.

3.2.2. Feature selection

Another, more indirect, aspect of duplicate presence involves feature selection. Typically, many classifiers suffer from the problem of the "curse of the dimensionality" and perform poorly if the dimensionality of the feature space is too high. And even if the feature-space size is not inherently a problem, some form of feature selection may be necessary to make learning with large datasets computationally feasible. Common feature-selection techniques used in text mining, such as χ^2 , mutual information or odds-ratio, are based upon estimating the asymmetry in the distribution of each term across the classes, a process which judges the most asymmetric features as most relevant [41]. Naturally, high duplicate rates involving certain features increase their chance of being selected, to the possible detriment of others.

3.2.3. Data sampling

High duplication rates may also affect the sampling regimens used for data collection, especially those based upon uniform sampling. Alternative sampling strategies, such as active sampling, might be effective in reducing duplicate presence in the leaning corpus.

4. Validation with unique data

Even though the presence of duplicates in the learning sample may be difficult to avoid, some of the potentially negative effects of their presence could be alleviated by using a (possibly much smaller) duplicate-free validation sample for the final tuning of a trained classifier (e.g., choosing the decision threshold) and/or for estimating its classification accuracy. Although the premise of this paper is that duplicates may be impossible to eliminate from large databases, the task becomes more feasible for smaller data collections (which makes it even possible to apply techniques based on pairwise document comparisons – see Section 7.1 for an overview). We believe that the bias caused by duplicate presence can be at least partially countered by tuning a trained classifier with a duplicate-free sample.

Note that the traditional techniques, such as cross-validation or bootstrap sampling, assume that the original learning collection represents an i.i.d. sample from the underlying distribution and attempt to closely replicate it for test/validation purposes. Research in the related problem of learning with imbalanced data typically uses ROC analysis for measuring classifier performance, since an ROC does not depend on the relative representation of the positive and negative examples in the test collection [28]. An ROC will, however, be influenced by the relative distribution of sub-categories within a class and also by the presence of duplicates.

Let us assume the availability of an evaluation sample $\mathcal{T}$ consisting of legitimate, $\mathcal{L}$, and spam, $\mathcal{S}$, subsets. Given that $\mathcal{T}$ may contain duplicates, let $\mathcal{T}_u$ denote the set of unique points within $\mathcal{T}$ (with the corresponding unique-message subsets of $\mathcal{L}_u$ and $\mathcal{S}_u$ for the two classes). The misclassification cost of a classifier F over $\mathcal{T}$ can be expressed as:

$$\begin{aligned} \mathfrak{L} &= \sum_{x\in\mathcal{S}_u} v(x)\left[F(x)=\mathtt{L}\right] + \mathit{cost}\sum_{x\in\mathcal{L}_u} v(x)\left[F(x)=\mathtt{S}\right] \\ &= \mathfrak{L}_u + \mathfrak{L}_d \end{aligned}$$

where

$$\mathfrak{L}_u = \sum_{x\in\mathcal{S}_u}\left[F(x)=\mathtt{L}\right] + \mathit{cost}\sum_{x\in\mathcal{L}_u}\left[F(x)=\mathtt{S}\right] \qquad (3)$$

and $v(x)$ is the volume of message x in the test sample. In this work, we will use $\mathfrak{L}_u$ as the basis for accuracy calculation. In this way, it is essentially assumed that any content region might potentially become a source of duplicates. Note that (3) does not remove other forms of sample-selection bias, however. For example, $\mathcal{L}$ may still consist largely of non-personal emails, since these were the easiest ones to collect. Thus certain areas of unique content may be sampled more densely than others. To counter this within-class imbalance between different sub-categories, one might purposefully increase the weight of document the rare categories [28] – a process that can be seen as controlled *duplication* of the evaluation data. Note that this technique requires the knowledge of sub-category labels.

In the following experimental section we will use (3) indirectly, i.e., as a basis for calculating the area under ROC (AUC) [4]. Let us define the false-positive (FP) and false-negative (FN) rates of classifier F with respect to the unique data as:

$$FP = \frac{\sum_{x\in\mathcal{S}_u}[F(x)=\mathtt{L}]}{|\mathcal{S}_u|}$$

and the false-negative rate of a classifier as

$$FN = \frac{\sum_{x \in \mathcal{L}_u} [F(x) = \mathtt{S}]}{|\mathcal{L}_u|}$$

and let $\pi = \mathcal{S}_u / \mathcal{T}_u$. In the above $|\mathcal{L}_u|$ and $|\mathcal{S}_u|$ denote the number of elements in $\mathcal{L}_u$ and $\mathcal{S}_u$, respectively. Then the misclassification cost of classifier F with respect to unique messages can be expressed as:

$$\mathfrak{L}_u = \pi \cdot FP + (1 - \pi) \cdot cost \cdot FN$$

The relationship between FP and FN for a given classifier is known as the Receiver Operating Characteristic (ROC)[14]. For classifiers returning a numeric score, different choices of the decision threshold will result in different points along the ROC curve, while for non-adjustable classifier the curve is created by joining the given (FP, FN) point with (0,0) in (1,1) in the ROC space (see for details). Given the target values of π and *cost*, one can adjust the operating point of the given classifier so that the misclassification cost is minimized. Recently, the area under the ROC curve (AUC) [4] has been used as single-value metric which captures the average accuracy of a classifier under all operating conditions. We have adopted AUC as the primary measure of classification accuracy in the experiments performed in this work.

5. Experimental Setup

5.1. The objectives

Given the discussion above, we set out to experimentally assess the practical impact of the presence of duplicates on the performance of text classifiers in the spam detection context. In particular, we were interested in evaluating the influence of the rate of duplication, as expressed by the repetition of a document in the learning sample. To this end, duplicates were created artificially by increasing the count of certain documents selected from the original collection.

For better clarity (and to reflect the typical case), duplication was only applied to the spam portion of the data. Although non-spams can also occur in volume, duplication of legitimate email is much more content dependent than is the case for spam, which can be treated more uniformly. Also, since legitimate bulk-mailers do not purposefully randomize their emails, finding duplicates in such data is likely to be much easier than in the case of spam, and hence the problem of duplicate contamination of the learning sample is mostly applicable to the spam portion of the collection. We focused on the perceived more common scenario, but are nevertheless planning to address more general distributions of duplicates among different classes in the future.

5.2. The email data set

The data collection consisted of $29,683$ legitimate and $28,442$ spam messages, with unknown number of duplicates, collected over a 3-month period. The collection process relied on a large number of volunteers, whose decision regarding the nature of each email was taken at face value (i.e., no further label verification was applied). The feature set was limited to the words present in the subject line and textual body parts of each message, where a word was defined as a sequence of ASCII characters delimited by punctuation or whitespace. All words were converted to lowercase, with the exception of all capital words that were counted twice (once as all lowercase and once as all caps). Otherwise, multiple occurrences of the same word were ignored. No header features or attributes tied to spam-filtering domain knowledge were used on this study[2].

One important consequence of the use of binary document representation in conjunction with feature selection is that, it provides a convenient platform for some rudimentary duplicate detection, whereby messages projected onto the same set of most relevant features can be considered duplicates of each other. Our past research indicates the effectiveness of such a technique in deduplicating large collections of web pages [8]. In this work, this approach was used to pre-screen the original dataset for potential duplicates and thus ensure that the only duplicates used were the ones created synthetically. To reduce the influence of potential duplicates already present in the data, the original collection was first tokenized and projected onto the set of $5,000$ highest ranking word features according to the

[2] A email message constitutes a combination of formatted (header) and freeform (body) text. Although classification accuracy can be increased by incorporating domain knowledge in the analysis of header features [32], only the textual components of each message represented by the message body and the subject line were taken into account in this work. We used the bag-of-words representation of each message, where only the presence or absence of a word was taken into account. Binary feature representations have been used quite extensively in text retrieval and categorization (e.g., a text classification study with Support Vector Machines [11] reported highest accuracy when using binary features rather than *tf-idf* representations), and carry the advantage of efficient implementation when the dimensionality of the feature space is high.

Mutual Information (MI) criterion [3]. For each class, in cases where multiple messages resulted in exactly the same feature vector, only one random message was retained, with the rest getting discarded. The resulting message collection (now containing $25,751$ legitimate and $17,609$ spam messages) was then split into a training and a test set, such that $2/3$ of data were used for training and the rest were used for testing.

To examine the effects of duplicate messages on classification performance, a 10% portion of the spam training set was randomly selected and messages within this subset had their multiplicity increased by 1, 5, 10, 50 and 100 fold. The duplication was realized via sampling with replacement where, at each episode, a message was selected at random from the working set and its copy was added to the working set. That way, the actual multiplicity varied from message to message, but the overall presence of messages in the oversampled subset was increased by the desired factor. This also reflected to some extent the apparent self-selection effect, where spam messages observed in high volume in a given time interval often arrive at an even higher relative volume in the following time interval. The process of randomly selecting a different 10% subset of the spam training data was repeated 10 times, and we report results averaged over the 10 trials.

To assess the impact on duplicates on different strategies of data collection that might occur in practice we considered two cases:

- One where the amplified portion of the training data was simply merged with the remaining portion. This case corresponds to dense sampling (e.g., one where a complete message stream is captured within a certain time window). Note that in this setup increasing levels of duplication were accompanied by increasing overall class imbalance of the learning sample.
- One where, after merging, the resulting set was uniformly down-sampled to its original size. This case corresponds to uniform sparse sampling (e.g., one where every Nth message in a data stream is captured). In this setup the data was class balanced, with duplicates "pushing out" some of the remaining data.

One might also envision various active-sampling regimens, where the choice about sampling a message is tied to the performance of a classifier (or a committee of classifiers) but we did not consider them here.

5.3. The classifiers

Research in text categorization suggests that, due to the inherently high dimensionality of the data, linear classifiers tend to be sufficient for this task and, in fact, often outperform nonlinear approaches. We considered two linear classifiers that scale well to massive datasets: multinomial Naive Bayes [23] (NB) and Perceptron with Margins (PAM) [24], which have been successfully applied to text categorization problems. The accuracy was measured in terms of the area under the ROC curve (AUC) so as to assess the general quality of the classifier without committing to any particular operating conditions.

The multinomial version of the Naive Bayes algorithm [25] was chosen since it tends to outperform other variants in text classification applications [13]. For each input document, x, containing terms $[x_1, .., x_D]$ with frequencies $[f_1, .., f_D]$ ($f_i \geq 0$), a Naive Bayes score was produced in the form of the log-odds ratio

$$F(x) = \log\left(\frac{P(\mathtt{L}|x)}{P(\mathtt{S}|x)}\right) = const + \sum f_i \cdot \log\left(\frac{P(x_i|\mathtt{L})}{P(x_i|\mathtt{S})}\right)$$

where the class conditional probability of feature x_i was computed as

$$\begin{aligned} P(x_i|\mathtt{L}) &= \frac{1+\sum_{x\in\mathcal{L}} f_i(x)}{D+\sum_j\sum_{x\in\mathcal{L}} f_j(x)} \\ P(x_i|\mathtt{S}) &= \frac{1+\sum_{x\in\mathcal{S}} f_i(x)}{D+\sum_j\sum_{x\in\mathcal{S}} f_j(x)} \end{aligned}$$

with $\mathcal{L}$ and $\mathcal{S}$ denoting the legitimate and spam subsets of the learning sample. In the above, $f_i(x)$ is the frequency of feature x_i in x and D is the dimensionality of the feature space (dictionary size).

In the case of PAM, the classifier output was generated as:

$$F(x) = \sum_i w_i \cdot x_i + b$$

with the weight vector, w, and the bias term, b, resulting from an iterative training procedure, where at each step a correction in the form of

$$\begin{aligned} w_i &\leftarrow w_i + \eta y x_i \\ b &\leftarrow b + \eta y R^2 \end{aligned}$$

[3] The Mutual Information (MI) criterion was defined as:

$$MI(t) = \sum_{t\in\{0,1\}} \sum_{c\in\{\mathtt{L},\mathtt{S}\}} P(t,c)\log\frac{P(t,c)}{P(t)\,P(c)} \quad (4)$$

where t denotes the term (i.e., feature), c is the class label and $P(t,c)$ is the joint probability of t and c cooccurring in a document collection; $P(t)$ is the probability of a document (regardless of class) containing term t and $P(c)$ is the probability of a document belonging to class c.

was made if

$$y\left(\sum_i w_i \cdot x_i + b\right) \leq \tau$$

with τ representing the classification margin, η the learning rate and y equal to +1 for $x \in \mathcal{L}$. and equal to -1 otherwise. R was set to $\max_x \|x\|$. Additionally, to insure convergence, the λ-trick [24] was used, whereby the for an N-element training set, each input vector as well as the weight vector had its dimensionality increased by N, and for the jth training point, the extension vectors was set to 0 apart from the jth position, where it was set to a positive value of λ.

For each version of the training set, a feature selection stage was performed, where the top 1,000 features were chosen according to the Mutual Information criterion. The messages consisting the training and test set were then projected on this feature set and the induced NB and PAM models were applied to the test set. In the case of PAM, the feature vectors were normalized to unit length and PAM was trained with even margins ($\tau = 0.001$) at the rate of $\eta = 0.01$. The λ-trick was used to account for the possibility of non linearly-separable cases, with λ set to 0.25. Each training episode proceeded till convergence or until 100 iterations were reached – whichever came sooner.

6. Results

Table 1 presents the AUC (averaged over 10 random trials) performance of the classifiers in the two sampling scenarios and for each fold of duplication. The results are also plotted in Figures 1 and 2 for better visualization. As can be seen, AUC decreases with the amount of duplicate contamination of the training data. For both classifiers, and for both types of sampling, the AUC metric for duplicate rates greater than 1 is significantly lower (according to a one-tailed t-test with $\alpha = 0.05$) than in the case of $rate = 1$. The decrease is rather gradual, however, indicating that, in the current setup, a much higher contamination level would be necessary to greatly impact either classifier, although the outcome might be different for other learners. One reason for this might be due to the fact that 10% of the spam sample chosen for duplicate generation (i.e., about 1,000 messages) perhaps captures much of the content variability for this class (spam tends to focus on the same narrow categories over and over again), given that sampling was carried over the set of unique messages. We suspect that the effect of duplicate contamination might be more dramatic if their bulk represented a narrow content category (a realistic scenario for dealing with such a sample would be when spam complaints were to be gathered over a short period of time, in which case they might correspond to the campaign of a particular spammer who managed to penetrate the system defenses).

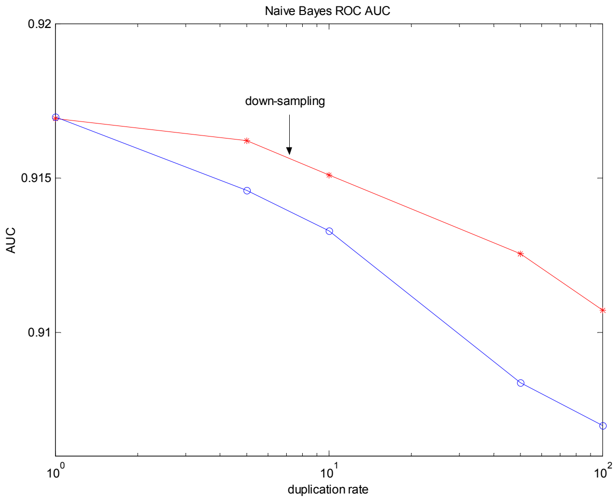

Figure 1. ROC AUC of Naive Bayes (avereged over 10 random trials). Duplication rate refers to randomly selecting a 10% portion of the spam training data and increasing their counts (on average) by the value of the rate shown. The graph indicated by an arrow corresponds to downsampling the spam portion of the training data so that the the training-set size remains constant.

Interestingly, for Naive Bayes, the down-sampling setup led to a better performance (significantly higher for folds 5, 50, 100 according to a one-tailed t-test with $\alpha = 0.05$). We believe that this could be attributed to the relatively high class-imbalance sensitivity of Naive Bayes, which also has been noted elsewhere [31]. One should be cautious when applying these results, however. Although not evident in the setup used in this work, one could easily envision a scenario where an extreme number of duplicates is present, in which case downsampling might result in compressing one side of the training set to copies of essentially one data point. In a more general context, the class-imbalance sensitivity of a given learner should be taken into account to decide what kind of sampling would be more beneficial (e.g., see [39]). In the case of PAM, on the other hand, down-sampling seems to somewhat degrade the performance, but this becomes significant only at the highest duplication rate of 100. Clearly, PAM appears to be not as sensitive as NB as far as type of sampling is concerned.

Although the experiments were not performed with the aim of comparing NB and PAM at the spam detection

Table 1. AUC results for for the Naive Bayes and PAM classifiers, averaged accross 10 random trials with standard deviations indicated. The experiments where duplicates increase the size of the training set are labeled as AUC-org, while the ones where the training set with duplicates is downsampled to its original size are labeled as AUC-down.

Duplication rate	AUC-org	AUC-down	AUC-org	AUC-down
	Naive Bayes		*Perceptron with margins*	
1	0.917±0.0011	0.917±0.0012	0.946±0.0014	0.946±0.0016
5	0.915±0.0019	0.916±0.0018	0.943±0.0011	0.943±0.0012
10	0.913±0.0030	0.915±0.0030	0.941±0.0014	0.940±0.0017
50	0.908±0.0050	0.913±0.0052	0.931±0.0029	0.930±0.0035
100	0.907±0.0035	0.911±0.0045	0.931±0.0026	0.925±0.0031

task, we note that at all duplication levels PAM scored significantly higher than NB on the dataset used. Actually, for $rate = 1$ a linear SVM performed only slightly better than PAM, resulting in AUC of 0.949.

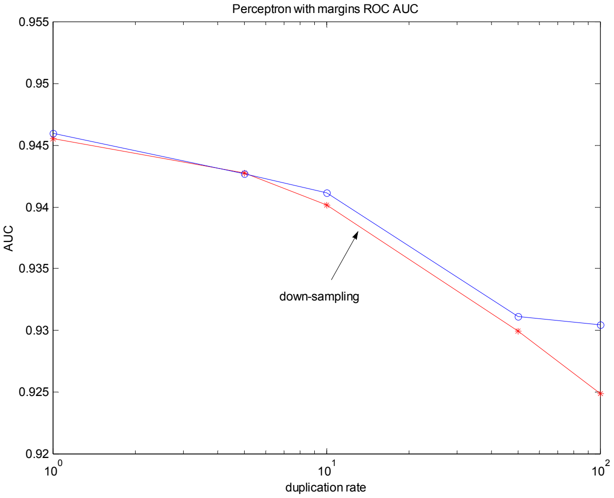

Figure 2. Classifcation accuracy of PAM in terms of ROC AUC (avereged over 10 random trials). Duplication rate refers to randomly selecting a 10% portion of the spam training data and increasing their counts (on average) by the value of the rate shown. The graph indicated by an arrow corresponds to down-sampling the spam portion of the training data so that the the training-set size remains constant. The two graphs differ significantly only at $rate =$ 100.

7. Prior and related work

7.1. Duplicate detection

The various approaches to finding these similar documents can be roughly classified as [17] *similarity based* and *fingerprint based.*

Approaches that compute document-to-document similarity measures [7][35][17] are similar to document clustering work [34] in that they use similarity computations to group potentially duplicate documents. In principle all pairs of documents need to be compared but in reality these approaches only evaluate documents with an overlap of terms. The basic hypothesis of similarity-based techniques is that similarity between different instances if the same document is higher than between unrelated documents.

Fingerprinting techniques examine a document as a stream of tokens [5][16][6]. The stream is broken into segments and each segment is hashed and referred to as a shingle. Each document is then represented as a set of shingles. The set of generated shingles or fingerprint is then compared against all other documents with matching shingles. To determine the similarity of two documents, a percentage of overlapping shingles is calculated. To combat the inherent efficiency issues, several optimization techniques were proposed to reduce the number of comparisons made [16][6].

The I-Match [8] approach eliminates the I/O costs by producing a single hash representation of a document and guaranteeing that a single document will map to one and only one cluster, while still providing fuzziness of non-exact matching. Each document is reduced to a feature vector and term collection statistics are used to produce a binary feature selection-filtering agent. The filtered feature vector is then hashed to a single value for all documents that produced the identical filtered feature vector, thus producing an efficient mechanism for duplicate detection.

Most technique assume that documents are duplicates when they contain highly similar text, which may be of limited validity in the spam filtering domain. Hall [15] examined the question of can duplicate detection approaches keep up with hostile environments and found that it is easier to mask messages then to determine they are the same, thus duplicate detection alone can not keep up with spam techniques.

7.2. Learning with imbalanced data

Dealing with problems where the learning sample contains much fewer examples of at least one the classes has been noticed to be an important problem in machine learning and data mining [18]. Due to the natural rarity of certain events [38], or due to the difficulty (or high cost) of sampling certain types of data [42], a learner if often faced with sample that is higly class-imbalanced. This often poses difficulties for inducing and measuring the accuracy of classifiers, especially since many standard techniques are geared towards minimizing the raw error rate. In some cases, an appropriate setting of the classifier's operating point may be sufficient [27], but published results (e.g., [39][20]) suggest that rebalancing of the learning sample (via downsampling or oversampling [22]) tends to be generally effective. Interestingly, there is evidence that optimum mixing proportions do not have to be equal and may be dataset and classifier dependent [39]. All in all, although the presence of class imbalance tends to be a challenge to classifiers, understanding of the nature of the problem remains incomplete and, indeed, there have been reports that for some tasks extreme class imbalance may in fact be beneficial [21].

7.3. Filtering of email spam

Spam filtering based on actual text of email messages can be seen as a special case of text categorization, with the categories being spam and non-spam. Although the task of document/text categorization has been researched extensively, its particular application to email data, and especially detection of spam, is relatively recent. Most researchers focus on creating personal categorizers/filters as opposed to system-wide solutions, which have to perform this function for a large and diversified group of users.

Cohen [10] considered the general problem of routing emails into a set of folders and demonstrated that an automatic rule learning system (RIPPER [9]) rivals hand-crafted rules, while being much more easy to maintain.

Provost [29] compared RIPPER with Naive Bayes (NB) in the email categorization task (in [29] spam filtering was treated as special case of categorization) and found NB to be more effective (Rennie [30] used NB to develop `ifile`, an email foldering system).

The first research studies which focused primarily on the problem of filtering spam were those of Sahami *et al.* [32] and Drucker *et al.* [11]. In [32], the authors applied NB to the problem of building a personal spam filter. NB was advocated due to its previously demonstrated robustness in the text-classification domain, and due to its ability to be easily implemented in a cost-sensitive decision framework.

The validity of SVMs' effectiveness in spam detection (suggested in [32]) was verified by Drucker *et al.* [11], who compared SVMs with RIPPER, a TF-IDF based classifier and a boosted ensemble of C4.5 trees.

In a series of papers, Androutsopoulos *et al.* [1][2][3] extended the NB filter proposed in [32], by investigating the effect of different numbers of features and training-set sizes on the filter's performance. The accuracy of the NB filter was shown to greatly outperform the keyword-based filter used by Outlook 2000 [2]. NB was also shown comparable with a memory-based classifier (k-nearest neighbor (k-nn))[3], with a combination of the two via stacking producing the best results [33].

8. Conclusions

Detection of duplicates in large data collections is an important problem in machine learning and data mining. The actual consequences of the presence of duplicates are less understood, however, and may be application dependent. In this work we examined the challenges and potential risks of training and evaluating classifiers using data contaminated with duplicates in text classification. Our focus was the task of spam detection, where varying rates of data duplication are common, but the results should be applicable to other domains.

We examined the practical impact of duplicates on the accuracy of classification (as measured on a duplicate-free collection) and its dependence on the rate of duplication. The results obtained indicate that the presence of duplicates in the learning sample does indeed pose a problem, even if they cover a diverse range of content. This stresses the importance of performing data deduplication which, even if not perfect, can at least insure that the level of duplicate contamination is reduced. This seems to be essential, since our results indicate that the loss of classification accuracy can be strongly correlated with the contamination level.

It is interesting that, even if generally negatively affected by duplicates in the learning sample, classifiers such as the Naive Bayes and PAM learners used in our study can be fairly robust, if the presence of duplicates is not too extreme. This is quite encouraging, given that duplicate detection tends to be imperfect, especially in domains such as spam filtering, where there is an active effort on the part of document creators (i.e., spammers) to avoid message duplication attempts. We expect, however, that the impact of duplicates is likely to very with relation to their distribution in the input (content) space, and also among different classes. We would like to investigate this further in the future.

References

[1] I. Androutsopoulos, J. Koutsias, K. Chandrinos, G. Paliouras, and C. Spyropoulos. An Evaluation of Naive Bayesian Anti-Spam Filtering. In G. Potamias, V. Moustakis, and M. van Someren, editors, *Proceedings of the Workshop on Machine Learning in the New Information Age: 11th European Conference on Machine Learning (ECML 2000)*, pages 9–17. 2000.

[2] I. Androutsopoulos, J. Koutsias, K. Chandrinos, G. Paliouras, and C. Spyropoulos. An Experimental Comparison of Naive Bayesian and Keyword-Based Anti-Spam Filtering with Encrypted Personal E-mail Messages. In N. Belkin, P. Ingwersen, and M. Leong, editors, *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000)*, pages 160–167. 2000.

[3] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. Spyropoulos, and P. Stamatopoulos. Learning to Filter Spam E-Mail: A Comparison of a Naive Bayesian and a Memory-Based Approach. In H. Zaragoza, P. Gallinari, and M. Rajman, editors, *Proceedings of the Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2000)*, pages 1–13. 2000.

[4] A. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[5] S. Brin, J. Davis, and H. Garcia-Molina. Copy detection mechanisms for digital documents. In *Proceeding of SIGMOD*, pages 398–409, 1995.

[6] Broder. On the resemblance and containment of documents. *SEQS: Sequences '91*, 1998.

[7] C. Buckley, C. Cardie, S. Mardisa, M. Mitra, D. Pierce, K. Wagstaff, and J. Walz. The smart/empire tipster ir system. In *TIPSTER Phase III Proceedings*. Morgan Kaufmann, 2000.

[8] A. Chowdhury, O. Frieder, D. A. Grossman, and M. C. McCabe. Collection statistics for fast duplicate document detection. *ACM Transactions on Information Systems*, 20(2):171–191, 2002.

[9] W. Cohen. Fast effective rule induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, 1995.

[10] W. Cohen. Learning Rules that Classify E-Mail. In *Proceedings of the 1996 AAAI Spring Symposium on Machine Learning in Information Access*, 1996.

[11] H. Drucker, D. Wu, and V. Vapnik. Support Vector Machines for Spam Categorization. *IEEE Transactions on Neural Networks*, 10(5):1048–1054, 1999.

[12] C. Elkan. The foundations of cost-sensitive learning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI'01)*, pages 973–978, 2001.

[13] S. Eyheramendy, D. Lewis, and D. Madigan. On the Naive Bayes model for text categorization. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.

[14] T. Fawcett. ROC graphs: Notes and practical considerations. Technical Report HPL-2003-4, HP Labs, 2003.

[15] R. J. Hall. A countermeasure to duplicate-detecting anti-spam techniques. Technical Report 99.9.1, AT&T Labs Research, 1999.

[16] N. Heintze. Scalable document fingerprinting. In *1996 USENIX Workshop on Electronic Commerce*, November 1996.

[17] T. C. Hoad and J. Zobel. Methods for identifying versioned and plagiarised documents. *Journal of the American Society for Information Science and Technology*, 2002.

[18] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5), 2002.

[19] A. Kołcz and J. Alspector. SVM-based filtering of e-mail spam with content-specific misclassification costs. In *Proceedings of the Workshop on Text Mining (TextDM'2001)*, 2001.

[20] A. Kołcz and J. Alspector. Asymmetric Missing-Data Problems: Overcoming the Lack of Negative Data in Preference Ranking. *Information Retrieval*, 5(1):5–40, 2002.

[21] A. Kowalczyk and B. Raskutti. One class SVM for yeast regulation prediction. *SIGKDD Explorations*, 4(2), 2002.

[22] M. Kubat and S. Matwin. Addressing the curse of imbalanced training sets: one-sided selection. In *Proc. 14th International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann, 1997.

[23] D. D. Lewis. Naive (Bayes) at forty: the independence assumption in information retrieval. In *Proceedings of the 10th European Conference on Machine Learning*, pages 4–15, 1998.

[24] Y. Li, H. Zaragoza, R. Herbrich, J. Shawe-Taylor, and J. Kandola. The perceptron algorithm with uneven margins. In *Proceedings of ICML 2002*, pages 379–386, 2002.

[25] A. McCallum and K. Nigam. A comparison of event models for Naive Bayes text classification. In *Proceedings of the AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[26] A. McCallum, K. Nigam, and L. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2000)*, 2000.

[27] F. Provost. Machine learning from imbalanced data sets 101. In *the AAAI Workshop on Learning from Imbalanced Data Sets*, 2000.

[28] F. Provost and T. Fawcett. Robust Classification for Imprecise Environments. *Machine Learning*, 42:203–231, 2001.

[29] J. Provost. Naive-Bayes vs. Rule-Learning in Classification of Email. Technical report, Dept. of Computer Sciences at the U. of Texas at Austin, 1999.

[30] J. Rennie. ifile: An Application of Machine Learning to E-mail Filtering. In *Proceedings of the KDD-2000 Workshop on Text Mining*, 2000.

[31] J. D. M. Rennie. Improving multi-class text classification with Naive Bayes. Technical Report AITR-2001-004, Massachusetts Institute of Technology, 2001.

[32] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A Bayesian Approach to Filtering Junk E-Mail. In *Proceedings of the AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[33] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. Spyropoulos, and P. Stamatopoulos. Stacking Classifiers for Anti-Spam Filtering of E-Mail. In L. Lee and D. Harman, editors, *Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing (EMNLP 2001)*, pages 44–50. Carnegie Mellon University, 2001.

[34] G. Salton, C. Yang, and A. Wong. A vector-space model for information retrieval. *Communications of the ACM, 18*, 1975.

[35] M. Sanderson. Duplicate detection in the Reuters collection. Technical Report TR-1997-5, Department of Computing Science, University of Glasgow, 1997.

[36] S. Sarawagi and A. Bhamidipaty. Interactive deduplication using active learning. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002)*, 2002.

[37] N. Shivakumar and H. Garcia-Molina. SCAM: A copy detection mechanism for digital documents. In *Proceedings of 2nd International Conference in Theory and Practice of Digital Libraries (DL'95) , Austin, Texas*, 1995.

[38] G. Weiss and H. Hirsh. Learning to predict extremely rare events. In *the AAAI Workshop on Learning from Imbalanced Data Sets*, 2000.

[39] G. Weiss and F. Provost. The effect of class distribution on classifier learning: An empirical study. Technical Report ML-TR-44, Department of Computer Science, Rutgers University, 2002.

[40] W. E. Winkler. The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Bureau of Census, Washington, DC, 1999.

[41] Y. Yang and J. P. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97)*, pages 412–420, 1997.

[42] B. Zadrozny and C. Elkan. Learning and Making Decisions When Costs and Probabilities are Both Unknown. Technical Report CS2001-0664, UCSD, 2001.