E-Mail Classification Based on NMF

Andreas G. K. Janecek[*] Wilfried N. Gansterer[*]

Abstract

The utilization of nonnegative matrix factorization (NMF) in the context of e-mail classification problems is investigated. In particular, it is of interest how the improved interpretability of the NMF factors due to the non-negativity constraints (which is of central importance in various problem settings) can be exploited specifically when classifying e-mail data. This problem context motivates, for example, a new approach for initializing the factors of the NMF. We evaluate this approach and show how approximation accuracy can be increased and/or computational effort can be reduced compared to initialization strategies suggested earlier. Beyond that, various classification methods based on the NMF are investigated. It turns out that they are not only competitive in terms of classification accuracy with state-of-the-art classifiers, but also provide advantages in terms of computational effort (especially for low-rank approximations).

1 Introduction

About a decade ago unsolicited bulk e-mail ("spam") started to become one of the biggest problems on the Internet. A vast amount of strategies and techniques were developed and employed to fight e-mail spam, but none of them can be considered a final solution. In recent years, phishing ("password fishing") has become a severe problem in addition to spam e-mail. In contrast to unsolicited but harmless spam e-mail, pihishing is an enormous threat for all big internet based commercial operations. The term covers various criminal activities which try to fraudulently acquire sensitive data or financial account credentials from internet users, such as account user names, passwords or credit card details. Phishing attacks use both social engineering and technical means.

Generally, e-mail classification methods can be categorized into three groups, according to their point of action in the e-mail transfer process. These three groups are pre-send methods, post-send methods and new protocols, which are based on modifying the transfer process itself. Most of the currently used e-mail filtering techniques belong to the group of post-send methods. Amongst others, this group comprises techniques such as black- and whitelisting or rule-based filters, that block e-mail depending on a pre-determined set of rules. These rules can also be thought of as features describing an e-mail message. After extracting the features, a classification process can be applied to predict the class (ham, spam, phishing) of unclassified e-mail. A popular method to increase the speed of the classification process is to perform feature subset selection (removal of redundant and irrelevant features) or dimensionality reduction (use of low-rank approximations of the original data) prior to the classification.

Low rank approximations – which are also utilized in other data mining applications such as image processing, drug discovery, or text mining – are used to either or both (*i*) reduce the required storage space, (*ii*) give more efficient representations of the relationship between data elements. Beside well known techniques like principal component analysis and singular value decomposition, there are several other methods for achieving this goal like vector quantization [21], factor analysis [10], QR-decomposition [9] or CUR-decomposition [5]. In recent years, another approximation technique for *nonnegative* data has been used successfully in various fields. The *nonnegative matrix factorization* (NMF, see Section 2) searches for reduced rank *nonnegative* factors $\mathbf{W}$ and $\mathbf{H}$, that approximate a given nonnegative data matrix $\mathbf{A}$, such that $\mathbf{A} \approx \mathbf{WH}$.

In this paper, we investigate the application of NMF to the task of e-mail classification. Motivated by this context, we investigate a new initialization technique for NMF based on ranking the original features in comparison to standard random initialization and other initialization techniques for NMF described in the literature. Our approach shows faster reduction of the approximation error than random initialization and comparable results to existing but sometimes more time-consuming approaches. Moreover, we analyze classification methods based on NMF. In particular, we introduce a new idea how NMF can be combined with LSI (latent semantic indexing) and compare it to standard LSI. Prior to that we take a short look at the interpretability of applying NMF to the context of e-mail classification. We try to take advantage of information provided by the basis vectors in $\mathbf{W}$ (basis e-mails or the basis features).

[*] University of Vienna - Research Lab Computational Technologies & Applications. Lenaugasse 2/8, 1080-Vienna, Austria. Mail to: `<firstname>.<lastname>@univie.ac.at`

1.1 Related Work. The utilization of low-rank approximations in the context of e-mail classification has been analyzed in [7]. In this work, LSI was applied successfully on both, pure textual features and features extracted by rule-based filtering systems. Especially the features from rule-based filters allowed for a strong reduction of the dimensionality without loosing significant accuracy in the classification process. In [6] a different technique was applied to classify e-mail – an enhanced self-learning variant of greylisting (temporarily rejection of e-mail messages) was combined with a reputation-based trust mechanism to provide time for separate feature extraction and classification. This architecture minimizes the workload on the client side and the results show very high spam classification rates. A comparison of the classification accuracy achieved with feature subset selection and low rank approximation based on PCA in the context of e-mail classification can be found in [11].

In 1994 Paatero et al. [16] published a *Nature* article on *positive matrix factorization*, but an article in the same journal five years later by Lee and Seung [14] achieved much more popularity and is known as a standard reference for nonnegative matrix factorization. The two NMF algorithms introduced in [14] – *multiplicative update algorithm* and *alternating least squares* [1, 15] – provide good baselines against which newer algorithms (e.g., the *gradient descent* algorithm) can be judged.

NMF Initialization. All algorithms for computing the NMF are iterative and depend on the initialization of $\mathbf{W}$ and $\mathbf{H}$. While the general goal – to establish initialization techniques and algorithms that lead to better overall error at convergence – is still an open issue, some initialization strategies can improve the NMF in terms of faster convergence and faster error reduction. Although the benefits of good NMF initialization techniques are well known in the literature, rather few algorithms for non-random initializations have been published so far.

Wild et al. [18, 19, 20] were among the first to investigate the initialization problem of NMF. They used spherical k-means clustering based on the centroid decomposition [4] to obtain a structured initialization for $\mathbf{W}$. More precisely, they partition the columns of $\mathbf{A}$ into k clusters and select the centroid vectors for each cluster to initialize the corresponding columns in $\mathbf{W}$. Their results show faster error reduction than random initialization, thus saving expensive NMF iterations. However, since this decomposition must run a clustering algorithm on the columns of $\mathbf{A}$ it is expensive as a preprocessing step (cf. [13]).

Langville et al. [13] also provided some new initialization ideas and compared the aforementioned centroid clustering approach and random seeding to four new initialization techniques. While two algorithms (Random Acol and Random C) only slightly decrease the number on NMF iterates and another algorithm (Co-occurence) turns out to contain very expensive computations, the *SVD-Centroid* algorithm clearly reduces the number of NMF iterations. The algorithm initializes $\mathbf{W}$ based on a SVD-centroid decomposition [18] of the low dimensional SVD factor $\mathbf{V}_{n\times k}$, which is much faster than a centroid-decomposition on $\mathbf{A}_{m\times n}$ since $\mathbf{V}$ is much smaller than $\mathbf{A}$. Nevertheless, the SVD factor $\mathbf{V}$ must be available for this algorithm, and the computation of $\mathbf{V}$ can time-consuming again.

Boutsidis et al. [2] initialized $\mathbf{W}$ and $\mathbf{H}$ using a technique called *Nonnegative Double Singular Value Decomposition* (NNDSVD) which is based on two SVD processes – one approximating the data matrix $\mathbf{A}$ (rank-k approximation), the other approximating positive sections of the resulting partial SVD factors. The authors performed various numerical experiments and showed that NNDSVD initialization is better than random initialization in term of faster convergence and error reduction in all test cases, and generally appears to be better than the centroid initialization in [18], except for one algorithm.

1.2 Synopsis. This paper is organized as follows: In Section 2 we review some basics of NMF and make some comments on the interpretability of the basis vectors in $\mathbf{W}$ in the context of e-mail classification (basis features and basis e-mails). We also provide some information about the data and feature sets used in this paper. Some ideas about new NMF initialization techniques are discussed in Section 3, and Section 4 focuses on new classification methods based on NMF. We conclude our work in Section 5.

2 Background

Before we make some remarks on the interpretability of the NMF factors $\mathbf{W}$ and $\mathbf{H}$ in the e-mail classification context, we briefly review the definition of NMF in the next paragraph, followed by a description of our data sets.

2.1 Nonnegative Matrix Factorization. The *nonnegative matrix factorization* (NMF) [16, 14] consists of reduced rank *nonnegative* factors $\mathbf{W} \in \mathbb{R}^{m\times k}$ and $\mathbf{H} \in \mathbb{R}^{k\times n}$ with (problem dependent) $k \ll min\{m, n\}$ that approximate a given nonnegative data matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{A} \approx \mathbf{WH}$.

The underlying non-linear optimization problem can generally be stated as

$$\min_{W,H} f(\mathbf{W},\mathbf{H}) = \frac{1}{2}||\mathbf{A} - \mathbf{WH}||_F^2$$

where $||.||_F$ is the Frobenius norm. Although the Frobenius norm is commonly used to measure the error between original data **A** and the low rank approximation **WH**, other measures are also possible [13, 14]. Due to its nonnegativity constraints, NMF produces so-called "additive parts-based" representations of the data [14] (in contrast to many other linear representations such as SVD, PCA or ICA). This makes the interpretation of the NMF factors much easier than for factors containing positive and negative entries.

Algorithms for Computing NMF. NMF algorithms can be divided into three general classes: multiplicative update (MU), alternating least squares (ALS) and gradient descent (GD) algorithms. A review of these three classes of algorithms can be found in [1]. In this paper, we use implementations of the MU and ALS algorithm (these algorithms do not depend on an step size parameter, as it is the case for GD) from the statistics toolbox (v6.2) in MATLAB. The termination criteria for both algorithms were also adapted from the MATLAB implementation.

2.2 Data Sets. The data sets used for evaluation consist of 15 000 e-mail messages, divided into three groups – ham, spam and phishing. The e-mail messages were taken partly from *Phishery*[1] and partly from the 2007 TREC corpus[2]. The e-mail messages are described by 133 features. A part of these features is purely text-based, other features comprise online features and features extracted by rule-based filters. Some of the features specifically test for spam messages, while other features specifically test for phishing messages. The structure of phishing messages tends to differ significantly from the structure of spam messages, but it may be quite close to the structure of regular ham messages (because for a phishing message it is particularly important to look like a regular message from a trustworthy source). A detailed discussion and evaluation of this feature set has been given in [8].

The e-mail corpus was split up into two sets (for training and for testing), one consisting of the oldest 4 000 e-mail messages of each class (12 000 messages overall), and one consisting of the newest 1 000 e-mail messages of each class (3 000 messages overall). This chronological ordering on historical data allows for simulating the changes and adaptations in spam and phishing messages which occur in practice. Both e-mail sets are ordered by the classes – the first group in each set consists of ham messages, followed by spam and phishing messages. Due to the nature of the features the data sets are rather sparse. The bigger (training) set has a sparsity (percentage of zero entries) of 84.7% and the smaller (testing) set has a sparsity of 85.5%. As preprocessing step we scaled all feature values to [0,1] to ensure that they have the same range.

2.3 Interpretation. A key characteristic of NMF is the ability to extract the underlying data as basis vectors in **W**. The second NMF factor **H** contains basis coefficients. With these coefficients the columns of **A** can be represented in the basis given by the columns of **W**. In the context of e-mail classification, **W** may contain *basis features* or *basis e-mails*, depending on the structure of the original data. If NMF is applied to an *e-mail* × *feature* matrix (i.e., every row in **A** corresponds to an e-mail message), then **W** contains k basis *features*. If NMF is applied on the transposed matrix (*feature* × *e-mail* matrix, i.e., every column in **A** corresponds to an e-mail message), then **W** contains k basis *e-mail messages*.

Basis Features. Figure 1 shows three basis features $\in \mathbb{R}^{12\,000}$ (for k=3) for our bigger data set when NMF is applied to an *e-mail* × *feature* matrix. The three different groups of objects – ham (first 4 000 messages), spam (middle 4 000 messages) and phishing (last 4 000 messages) – are easy to identify. The group of phishing e-mail tends to yield high values of basis feature 1, while basis feature 2 shows the highest values for the spam messages. The values of basis feature 3 are generally smaller than those of basis features 1 and 2, and this basis feature is clearly dominated by the ham messages.

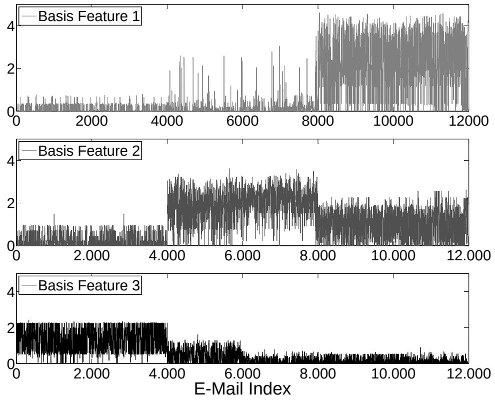

Figure 1: Basis features for $k = 3$

[1] http://phishery.internetdefence.net/

[2] http://trec.nist.gov/data/spam.html

Basis E-Mail Messages. The three basis e-mail messages $\in \mathbb{R}^{133}$ (again for k=3) resulting from NMF on the transposed (*feature* × *e-mail*) matrix are plotted in Figure 2. The figure shows two features (#16 and #102) that have a high value in all basis e-mails, indicating that these features do not distinguish well between the three classes of e-mail. Other features show better distinction between classes, for example the features 89-91 and 128-130 have a high value in basis e-mail 1, and are (close to) zero in the other two basis e-mails. Investigation of the original data shows that these features tend to have high values for phishing e-mail – indicating that the first basis e-mail represents a phishing message. Using the same procedure, the third basis e-mail can be identified to represent ham messages (indicated by features 100 and 101). Finally, basis e-mail 2 represents spam.

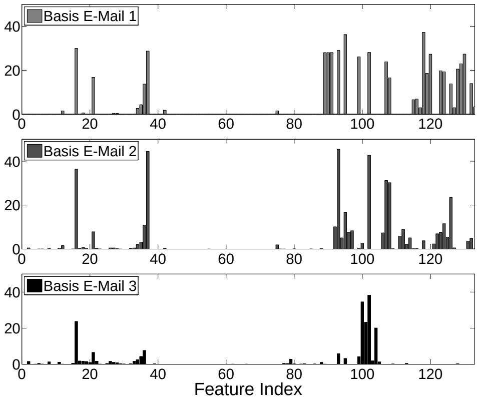

Figure 2: Basis e-mail messages for $k = 3$

This rich structure observed in the basis vectors should be exploited it in the context of classification methods. However, the structure of the basis vectors heavily depends on the concrete feature set used. Thorough investigation of this aspect is one of the topics of our ongoing work.

3 NMF Initialization Based on Feature Ranking

As already mentioned in the Section 1.1, initialization of NMF is an important issue to speed-up convergence and reduce the error in NMF algorithms. Although the benefits of good initialization are well known, randomized seeding of **W** and **H** is still the standard approach for many NMF algorithms. Existing approaches such as the initializations based on spherical k-means clustering [18] or nonnegative double singular value decomposition (NNDSVD) [2] can be rather time consuming. Obviously, the trade-off between computational cost in the initialization step and the computational cost the actual NMF algorithm needs to be chosen carefully. In some situations, an expensive preprocessing step may overwhelm the cost savings in the later applied NMF update steps. In the following, we introduce a simple and fast initialization step based on feature subset selection and show comparisons with random initialization and the NNDSVD approach mentioned before.

3.1 Feature Subset Selection. The main idea of feature subset selection (FS) is to rank features according to how well they differentiate between object classes. Redundant or irrelevant features can then be removed from the data set as they can lead to a reduction of the classification accuracy or clustering quality and to an unnecessary increase of computational cost. The output of the FS process is a ranking of features based on the applied FS algorithm. The two feature subset selection methods used in this paper are *information gain* and *gain ratio*, both reviewed briefly in the following.

Information Gain. One option for ranking the features of e-mail messages according to how well they differentiate the three classes ham, spam, and phishing is to use their *information gain*, defined as $\text{gain}(X, C) := \text{info}(C) - \text{info}_x(C)$ for a set of class labels C and a feature set X. The function info(C) is Shannon's entropy function and info_x is the conditional entropy function defined as $\text{info}_x(C) := \sum_{v \in X} P(v) * P(C|v)$, where $P(v)$ is the probability of v and $P(C|v)$ the conditional probability of C given v.

Gain Ratio. Since the information gain defined before favors features which assume many *different* values, we also ranked the features based on their *information gain ratio* $\text{gainratio}(X, C) := \text{gain}(X, C)/\text{splitinfo}(X)$ with $\text{splitinfo}(X) := -\sum_{v \in X} P(v) * \log_2 P(v)$.

3.2 FS-Initialization. After determining the feature ranking based on information gain and gain ratio, we use the k first ranked features to initialize **W** (denoted as *FS-initialization* in the following). Since FS aims to reduce the *feature* space, our initialization is motivated by the perspective that **W** contains basis features (i.e., every row in **A** corresponds to an e-mail message, cf. Section 2.3). FS methods are usually very fast (see, for example, [11] for a comparison of information gain and PCA runtimes). This fast and straightforward procedure can be used as a computationally cheap but effective initialization step. A detailed runtime comparison of information gain, gain ratio, NNDSVD, random seeding and other initialization methods is part of our ongoing work, as well as theinitialization of **H** – at the moment **H** is randomly seeded.

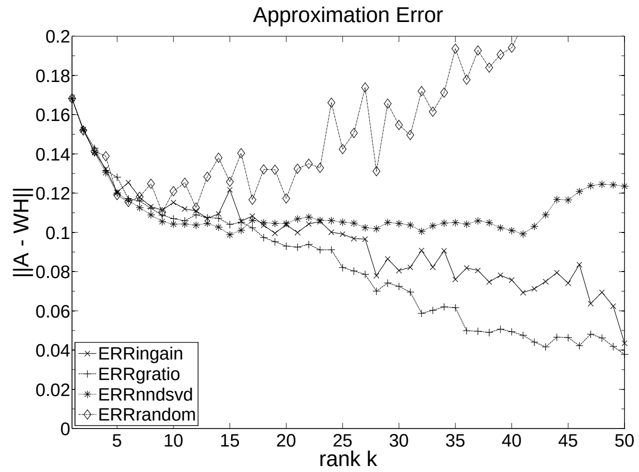

Figure 3: Approximation error for different values of rank k using the ALS algorithm (*maxiter*=5)

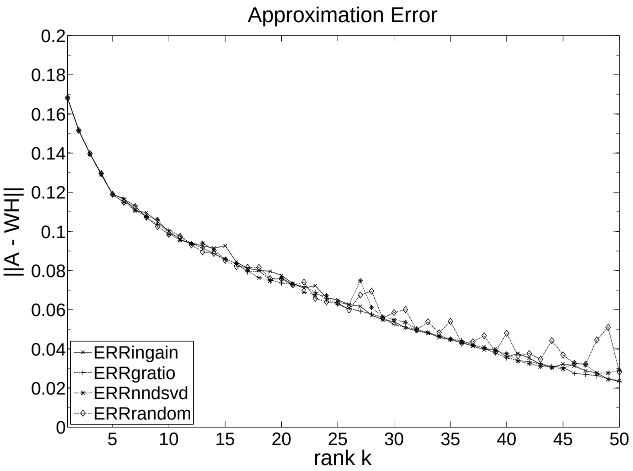

Figure 4: Approximation error for different values of rank k using the ALS algorithm (*maxiter*=30)

Results. Figures 3 and 4 show the NMF approximation error for our new initialization strategy (for both information gain (ERRingain) and gain ratio (ERRgratio) feature ranking) as well as for NNDSVD (ERRnndsvd) and random initialization (ERRrandom) when using the ALS algorithm. Note that when the maximum number of iterations inside the NMF (*maxiter*) is high (see Figure 4, *maxiter*=30), the approximation errors are very similar for all initialization strategies used. Contrary to that, with a small number of iterations (see Figure 3, *maxiter*=5), it is clearly visible that random seeding cannot compete with NNDSVD and FS-initialization. Moreover, for this small *maxiter*, the FS-initializations (for both information gain and gain ratio ranking) show better error reduction than NNDSVD with increasing rank of k. For higher *maxiter* the gap decreases until the error curves become basically identical when *maxiter* is about 30 (see Figure 4).

Runtime. Figure 5 shows the runtimes for computing NMF for different values of rank k and different values of *maxiter* using the ALS algorithm. The algorithms terminated when the number of iterations exceeded the pre-defined threshold *maxiter*, i.e., the approximation error was not integrated in the stopping criterion. Consequently, the runtimes do not depend on the initialization strategy used (neglecting the marginal runtime savings due to sparse initializations). In this setup, a linear relationship between runtime and rank k can be observed, as shown in Figure 5. It is clearly illustrated that reducing the number of iterations (lower values of *maxiter*) brings important reductions in runtimes. This underlines the benefits of our new initialization techniques: As Figure 3 has shown, our FS-initialization reduces the approximation error compared to existing approaches. Table 1 compares runtimes needed to achieve different approximation error thresholds with different values of *maxiter* for our IG-initialization. Obviously, a given approximation error $||\mathbf{A} - \mathbf{WH}||$ can be achieved much faster with small *maxiter* and high rank k than with high *maxiter* and small rank k.

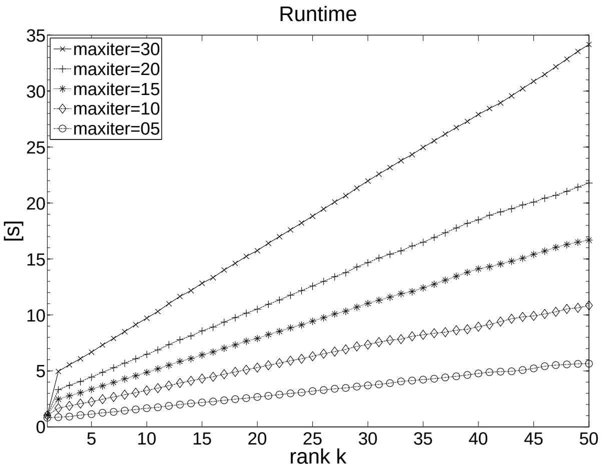

Figure 5: NMF runtime measurements (ALS)

$\|\|A - WH\|\|$	*maxiter*=05	*maxiter*=30
0.10	2.37s (k=17)	10.31s (k=11)
0.08	3.41s (k=27)	15.23s (k=19)
0.06	3.91s (k=32)	20.09s (k=27)
0.04	5.63s (k=49)	26.74s (k=38)

Table 1: Runtine comparison for information gain initialization for different values of *maxiter*

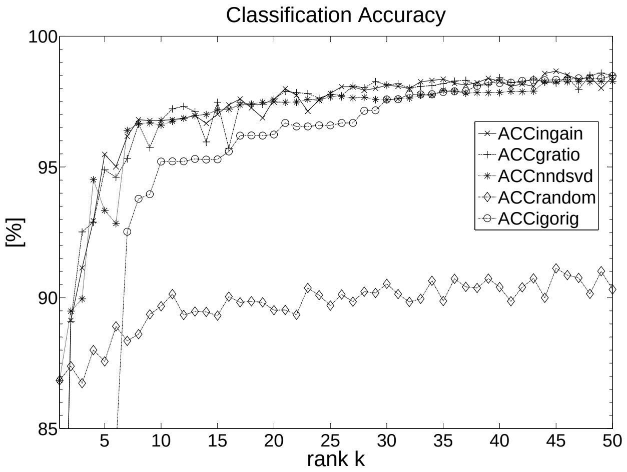

Figure 6: SVM (RBF-kernel) classification accuracy for different initialization methods using the MU algorithm (*maxiter*=5)

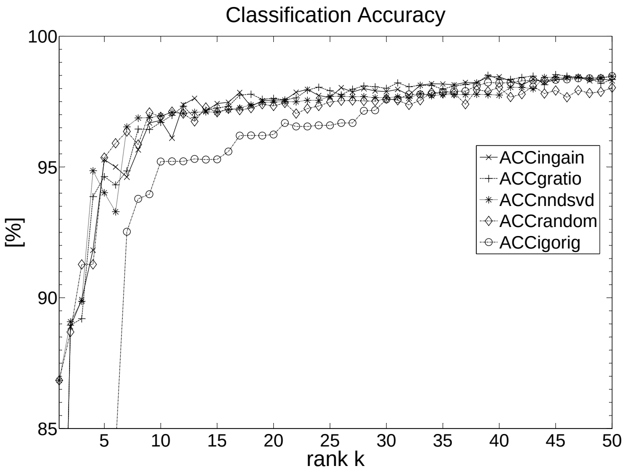

Figure 7: SVM (RBF-kernel) classification accuracy for different initialization methods using the MU algorithm (*maxiter*=30)

4 NMF-based Classification Methods

In this section we investigate various classification algorithms which utilize the NMF for developing a classification model. First, we look at the classification accuracy achieved with the basis features in **W** when initialized with the techniques explained in Section 3. Since in this case, the NMF is computed on the complete data, this technique can only be applied on data that is already available before the classification model is built.

In the second part of this section we introduce a classifier based on NMF which can be applied dynamically to new e-mail data. We present a combination of NMF with an LSI approach and show first comparisons with standard LSI (based on SVD).

4.1 Classification using Basis Features. Figures 6 and 7 show the overall classification accuracy for a ternary classification problem (ham, spam, phishing) using different values of *maxiter* for all four initialization strategies mentioned in Section 3. As classification algorithm we used a support vector machine (SVM) with a radial basis kernel provided by the MATLAB LIBSVM (version 2.88) interface [3]. For the results shown in this section, we applied a 5-fold cross validation on the larger e-mail corpus (consisting of 12 000 e-mail messages, cf. Section 2.2).

The results based on the four NMF initialization techniques (ACCingain, ACCgratio, ACCnndsvd and ACCrandom) were achieved by applying a SVM on the rows of **W**, where every e-mail message is described by k basis *features* (cf. Section 2.3). As NMF algorithm we used multiplicative update (MU). For comparison with original features, we applied a standard SVM classification on the e-mail messages characterized by k best ranked information gain features (ACCigorig). The graph for ACCigorig is identical in both figures since the *maxiter* factor in the NMF algorithm has no influence on the result.

Classification Results. For rank $k < 30$, the ACCigorig results are markedly below the results achieved with non-randomly initialized NMF (ACCingain, ACCgratio and ACCnndsvd). This is not very surprising, since **W** contains compressed information about all features (even for small ranks of k). It is very interesting to notice the low classification accuracy on **W** based on a random NMF initialization (ACCrandom) for *maxiter*=5 (see Figure 6). The classification result remains unsatisfactory even for large values of k. With increasing *maxiter* (cf. Figure 7), the classification accuracy for randomly seeded **W** increases and achieves results comparable to ACCingain, ACCgratio and ACCnndsvd. Comparing the results of the FS-initialization and NNDSVD initialization it can be seen that there is no big gap in the classification accuracy. It is interesting to notice the small decline in the classification accuracy when k=6 for ACCnndsvd (in both figures). Surprisingly, the classification results for *maxiter*=5 are only slightly worse than for *maxiter*=30 – which is contrary to the approximation error results shown in Section 3. Consequently, a fast classification process is possible for small *maxiter* and small k (for example, the average classification accuracy over ACCingain, ACCgratio and ACCnndsvd is 96.75% for k=10 and *maxiter*=5, compared to 98.34% for k=50, *maxiter*=50).

4.2 Generalizing LSI Based on NMF.

Now we take a look at the classification process in a dynamic setting where newly arriving e-mail messages are to be classified. Obviously, it is not suitable to compute a new NMF for every new incoming e-mail message. Instead, a classifier is trained by applying NMF on a training sample and using the information provided in the factors **W** and **H** for classifying new data. In the following, we present adaptions of LSI based on NMF and compare them with standard LSI (based on SVD). Please note that in this section our data sets are transposed compared to the experiments in Sections 3 and 4.1. Thus, every column of **A** corresponds to an e-mail message.

Review of VSM and Standard LSI. A vector space model (VSM, [17]) is a widely used algebraic model where objects and queries are represented as vectors in a potentially very high dimensional metric vector space. Generally speaking, given a query vector **q**, the distances of **q** to all objects in a given *feature* × *object* matrix **A** can be measured (for example) in terms of the cosines of the angles between **q** and the columns of **A**. The cosine φ_i of the angle between **q** and the i-th column of **A** can therefore be computed as

$$(VSM) : cos\varphi_i = \frac{e_i^\top A^\top q}{||Ae_i||_2||q||_2}$$

Latent semantic indexing (LSI, [12]) is a variant of the basic vector space model. Instead of the original matrix **A**, the singular value decomposition (SVD) is used to construct a low rank approximation $\mathbf{A}_k$, such that $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top \approx \mathbf{U}_k\Sigma_k\mathbf{V}_k^\top =: \mathbf{A}_k$. When **A** is replaced with $\mathbf{A}_k$, then the cosine φ_i of the angle between **q** and the i-th column of **A** can be approximated as

$$(LSI) : cos\varphi_i \approx \frac{e_i^\top V_k\Sigma_k U_k^\top q}{||U_k\Sigma_k V_k^\top e_i||_2||q||_2}$$

Since some parts of the right side of this equation only need to be computed once ($\mathbf{e}_i^\top\mathbf{V}_k\Sigma_k$ and $||\mathbf{U}_k\Sigma_k\mathbf{V}_k^\top\mathbf{e}_i||_2$), LSI saves storage and computational cost. Besides that, the approximated data often gives a cleaner and more efficient representation of the relationship between data elements [13] and is able to uncover *latent* information in the data.

NMF-based Classifiers. We investigate two concepts for using NMF as low rank approximation within LSI (see Figure 8). The first approach simply replaces the approximation matrix within LSI with a different approximation matrix. Instead of $\mathbf{A} \approx \mathbf{U}_k\Sigma_k\mathbf{V}_k^\top$, we approximate **A** with another $\mathbf{A}_k = \mathbf{W}_k\mathbf{H}_k$. Note that when using NMF the value of k must be known prior to the computation of **W** and **H**. The cosine of the angle between **q** and the i-th column of **A** can then be approximated as

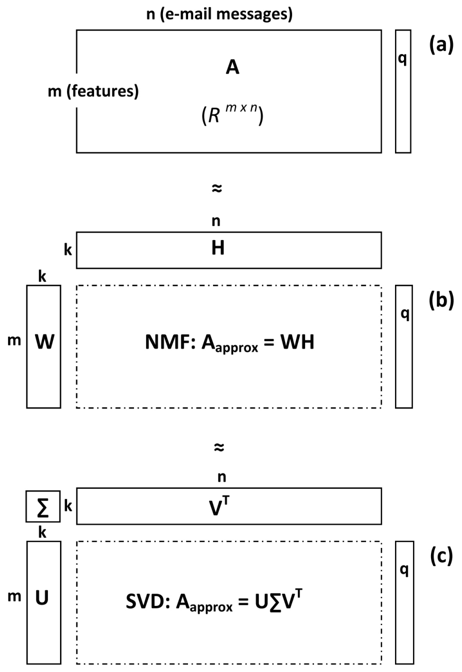

Figure 8: Overview - (a) basic VSM, (b) LSI using NMF, (c) LSI using SVD

$$(NMF\_1) : cos\varphi_i \approx \frac{e_i^\top H_k^\top W_k^\top q}{||W_k H_k e_i||_2||q||_2}$$

To save computational cost, the left part of the denominator can be computed a priori and the right part of the numerator ($\mathbf{W}_k^\top q$) can be computed before multiplying with $\mathbf{H}_k$.

Our second approach is based on the idea that the basis coefficients in **H** can be used to classify new e-mail. These coefficients are representations of the columns of **A** in the basis given by **W**. If **W**, **H** and **q** are given, we can calculate a column vector **x**, that minimizes the equation

$$\min_x ||Wx - q||.$$

Since **x** is the best representation of **q** in the basis given by **W**, we search for the closest column of **H** to assign **q** to one of the three classes of e-mail. Moreover, the error in the equation above indicates how close **q** is from the e-mail messages in **A**. The cosine of the angle between **q** and the i-th column of **A** can be approximated as

$$(NMF\_2) : cos\varphi_i \approx \frac{e_i^\top H^\top x}{||He_i||_2 ||x||_2}$$

It is obvious that the computation of the cosines is faster than for both other LSI variants mentioned before (since usually $\mathbf{H} \ll \mathbf{A}$), but the computation of $\mathbf{x}$ causes additional cost. These timing issues will be further investigated in future work.

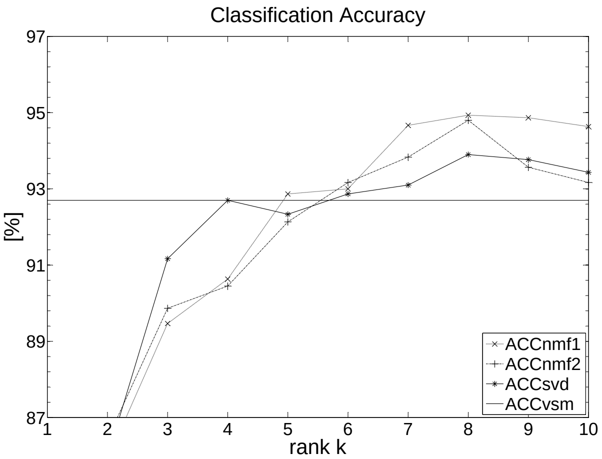

Figure 9: Classification accuracy for different values of rank k using different variants of LSI.

Classification Results. A comparison of the results achieved with the three LSI variants (ACCnmf1, ACCnmf2 and ACCsvd) and a basic VSM (ACCvsm) is shown in Figure 9. In contrast to Section 4.1, where we performed a cross validation on the bigger e-mail corpus, here we used the big corpus as training set and tested with the smaller corpus consisting of the 1 000 *newest* e-mail messages of each class. For classification, we considered the column of $\mathbf{A}$ with the smallest angle to $\mathbf{q}$ (no majority count) to assign $\mathbf{q}$ to one of the classes ham, spam and phishing. Overall, the classification results are good and very stable. For $k > 5$, all three LSI variants achieved better classification accuracy than the basic vector space model with all original features. Both NMF approaches (using the ALS algorithm with random initialization) show comparable and often even better results than standard LSI. Note that this improvement of a few percent is substantial in the context of e-mail classification. Moreover, the purely nonnegative linear representation within NMF make the interpretation of the NMF factors much easier than for standard LSI. At the moment our classification results were achieved using random initialization for NMF – an investigation of the classification accuracy achieved with initialization techniques from Section 3 is topic of our ongoing work.

5 Conclusion

The application of nonnegative matrix factorization (NMF) to ternary e-mail classification tasks (ham vs. spam vs. phishing messages) has been investigated. We have introduced a fast initialization technique based on feature subset selection (FS-initialization) which significantly reduces the approximation error when computing the NMF compared to randomized seeding of the NMF factors $\mathbf{W}$ and $\mathbf{H}$. Comparison of our approach with existing initialization strategies such as NNDSVD [2] shows basically the same accuracy when many NMF iterations are performed, and much better accuracy when the NMF algorithm is restricted to a small number of iterations.

Moreover, we proposed new classification methods which are based on NMF. We showed that using the basis features of $\mathbf{W}$ generally achieves much better results than methods using the original features. While the maximum number of iterations in the iterative process for computing the NMF seems to be a crucial factor for the classification accuracy based on NMF with random initialization, the classification results achieved with FS-initialization and NNDSVD only depend weakly on this parameter (see Figures 6 and 7). This is in contrast to the approximation error illustrated in Figures 3 and 4.

Finally, we constructed NMF-based classifiers to be applied on newly arriving e-mail without recomputing the NMF. For this purpose, we introduced two LSI classifiers based on NMF (computed with the ALS algorithm) and compared them to standard LSI based on singular value decomposition. Both new variants achieved a classification accuracy comparable with standard LSI. For a rank $k > 5$, the results are even better than for a standard VSM (see Figure 9).

Future Work. Our investigations provide several important and interesting directions for future work. First of all, we will set the focus on analyzing the computational cost of the initialization strategies (FS-initialization vs. NNDSVD) and the LSI variants introduced in Section 4 (standard LSI vs. NMF-based LSI). Moreover, we will look at updating schemes for our NMF-based LSI approach, since for real-time e-mail classification a dynamical adaptation of the training data (i.e., adding new e-mail to the training set) is essential. The initialization of the LSI variants based on NMF is also an important issue – currently the NMF factors are initialized randomly. We also plan to work on the initialization of $\mathbf{H}$ (meanwhile $\mathbf{H}$ is randomly initialized) for our FS-initialization (Section 3) and the comparison of the MU and ALS algorithms with other NMF algorithms (gradient descent, algorithms with sparseness constraints, etc.).

Acknowledgments. We gratefully acknowledge financial support from the CPAMMS-Project (grant# FS397001) in the Research Focus Area "Computational Science" of the University of Vienna. We also thank David Poelz for providing us with detailed information about the characteristics of the e-mail features.

References

[1] M. W. Berry, M. Browne, A. N. Langville, P. V. Pauca, and R. J. Plemmons, *Algorithms and applications for approximate nonnegative matrix factorization*, Computational Statistics & Data Analysis, 52 (2007), pp. 155–173.

[2] C. Boutsidis and E. Gallopoulos, *Svd based initialization: A head start for nonnegative matrix factorization*, Pattern Recogn., 41 (2008), pp. 1350–1362.

[3] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[4] I. S. Dhillon and D. S. Modha, *Concept decompositions for large sparse text data using clustering*, Machine Learning, 42 (2001), pp. 143–175.

[5] P. Drineas, R. Kannan, M. W. Mahoney, and L. A, *Fast monte carlo algorithms for matrices iii: Computing a compressed approximate matrix decomposition*, SIAM Journal on Computing, 36 (2004), pp. 184–206.

[6] W. N. Gansterer, A. Janecek, and K.-A. Kumer, *Multi-level reputation-based greylisting*, in Third International Conference on Availability, Reliability and Security (ARES 2008), Barcelona, Spain, 2008, IEEE Computer Society, pp. 10–17.

[7] W. N. Gansterer, A. Janecek, and R. Neumayer, *Spam filtering based on latent semantic indexing*, in Survery of Text Mining 2, vol. 2, Springer, 2008, pp. 165–183.

[8] W. N. Gansterer and D. Poelz, *E-mail classification for phishing defense*, to appear in Proceedings of ECIR 2009.

[9] G. H. Golub and C. F. Van Loan, *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)*, The Johns Hopkins University Press, October 1996.

[10] R. L. Gorsuch, *Factor Analysis*, Lawrence Erlbaum, 2nd ed., 1983.

[11] A. Janecek and W. N. Gansterer, *On the relationship between feature selection and classification accuracy*, JMLR: Workshop and Conference Proceedings, 4 (2008), pp. 90–105.

[12] A. N. Langville, *The linear algebra behind search engines*, in Journal of Online Mathematics and its Applications (JOMA), 2005, Online Module, 2005.

[13] A. N. Langville, C. D. Meyer, and R. Albright, *Initializations for the nonnegative matrix factorization*, in Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006.

[14] D. D. Lee and H. S. Seung, *Learning the parts of objects by non-negative matrix factorization.*, Nature, 401 (1999), pp. 788–791.

[15] ——, *Algorithms for non-negative matrix factorization*, Advances in Neural Information Processing Systems, 13 (2001), pp. 556–562.

[16] P. Paatero and U. Tapper, *Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values*, Environmetrics, 5 (1994), pp. 111–126.

[17] V. V. Raghavan and S. K. M. Wong, *A critical analysis of vector space model for information retrieval*, Journal of the American Society for Information Science, 37 (1999), pp. 279–287.

[18] S. M. Wild, *Seeding non-negative matrix factorization with the spherical k-means clustering*, Master's Thesis, University of Colorado, (2002).

[19] S. M. Wild, J. H. Curry, and A. Dougherty, *Motivating non-negative matrix factorizations*, in Proceedings of the Eighth SIAM Conference on Applied Linear Algebra, July 2003.

[20] ——, *Improving non-negative matrix factorizations through structured initialization*, Pattern Recognition, 37 (2004), pp. 2217–2232.

[21] A. B. Y. Linde and R. M. Gray, *An algorithm for vector quantizer design*, IEEE Transactions on Communications, (1980), pp. 702–710.