On the Effectiveness of Non-negative Matrix Factorization for Text Open-set Recognition

Angelo Impedovo[0000−0001−6837−3258] and Giuseppe Rizzo[0000−0001−9609−1741]

Niuma s.r.l.
Via Giacomo Peroni, 400, 00131 Rome, Italy
{angelo.impedovo, giuseppe.rizzo}@niuma.it

Abstract. Open-set recognition (OSR) is a more realistic approach than traditional multiclass classification in many real-world scenarios where an unforeseeable number of classes may arise at inference time. Specifically, OSR aims to recognize whether an instance belongs to one of the classes used for the training or not. In case it is, the instance is also categorized accordingly. On the contrary, when not recognized, OSR labels it with a generic class label indicating the rest of the world. Similarly to text classification, OSR approaches suffer from the curse of dimensionality and feature reduction could be beneficial. In this paper, inspired by previous evidence on text classification, we claim that Non-negative Matrix Factorization (NMF) of the tf-idf term-document matrix can also improve OSR accuracy on text data. Preliminary results on benchmark datasets prove our claim is correct and paves the way for future developments.

Keywords: Open-set Recognition · Non-negative Matrix Factorization · Machine Learning.

1 Introduction

The *Open-set recognition* (OSR) task extends traditional multiclass classification. As the name suggests, OSR aims to recognize whether a given test instance belongs to one of the classes used for the training and categorize it accordingly. On the contrary, in cases where it is not recognized, OSR labels it with a generic class label indicating the rest of the world. In a real-world scenario, where instances belonging to an unforeseeable growing number of possible classes are expected at inference time, gathering training instances for all of them may be impractical or even impossible, thus making the training of a multiclass classifier infeasible. Furthermore, traditional multiclass classification assumes that instances observed at inference time belong to the same classes seen at training time. Therefore, in such cases, OSR is a favorable solution. Figure 1 gives an intuitive idea of how OSR differs from ordinary classification by comparing their decision boundaries: while in traditional classification, a model separates classes with an open hyperplane, in OSR the goal is to find a tight margin around known-class training instances, so that those far away from the boundaries can be assigned to the generic class denoting the rest of the world. Therefore, adopt-

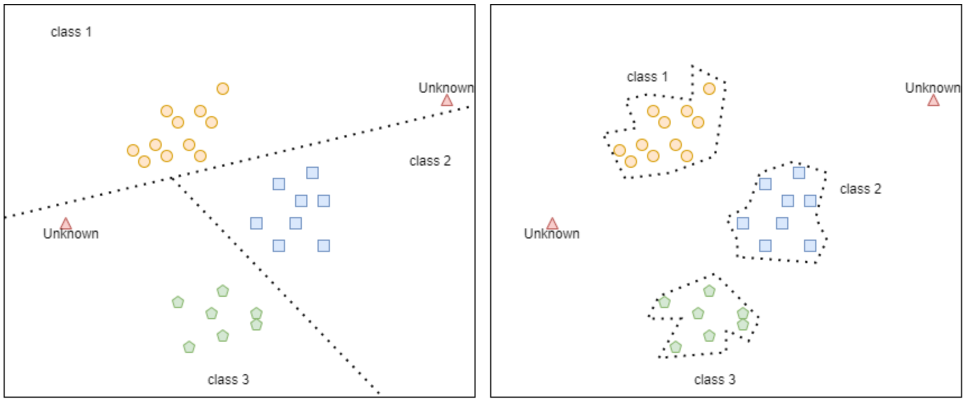

Fig. 1: 2D decision boundary of a multiclass classifier (left) and OSR (right)

ing OSRs is essential for the practical deployment of the model that should be able to accurately classify known-class instances while also effectively dealing with unknown-class ones.

Similarly to multiclass classification, OSR can be applied to a variety of data types, including high-dimensional data such as text. As such, OSR is subject to the *curse of dimensionality*, a problem arising when the number of dimensions (features) in a dataset increases. As the data dimensionality increases, the volume of the space overgrows so that the available data become sparse, making it difficult to catch statistically significant regularities in the data. To mitigate this issue, dimensionality reduction techniques aim to map high-dimensional data to a lower-dimensional space while preserving the nature of the original data.

Non-negative matrix factorization (NMF) has become a popular dimensionality reduction technique in recent years due to its parts-based, non-subtractive interpretation of the learned basis. Informally, NMF builds a number of features, each of them representing a part of the original data that can be combined to recreate it (e.g., in the case of a face image, such parts can correspond to "nose", "eyes" and so on). Given a non-negative data matrix V, NMF decomposes it into two non-negative matrices, W and H, such that $V = WH$. NMF has been effectively used to factorize tf-idf weighted term-document matrices, prior to text classification. In so doing, NMF analyzes the conceptual semantic space and, consequently, reduces the dimensionality of tf-idf document vectors, ultimately representing texts in terms of conceptual features.

In this paper, our main claim is that NMF approaches can improve open-set recognition accuracy. Although previous studies [15] have shown that the local conceptual semantic space generated based on NMF can result in better text classification accuracy, as far as we can tell, there is no evidence that such a result also holds in the open-set recognition scenario. Specifically, since OSRs are trained on instances belonging to known classes and tested on both known and unknown classes, we claim that conceptual features learned from NMF on known-class instances can significantly boost the recognition accuracy, even on unforeseen classes. Guided by previous studies on text classification, we design experiments on open-set recognition data to test this claim. The remainder of this paper is organized as follows: Section 2 introduces the recent literature on open set recognition and non-negative matrix factorization techniques and Section 3 introduces the basics notions. Then, Section 4 discusses the experimental results, before concluding the manuscript in Section 5.

2 Related Works

In recent years, the open-set recognition problem gained momentum in both the industrial community, which developed new OSR methodologies, and the research community, which in its turn considered multiple applications of OSR to real-world problems. This section recalls the main frameworks and applications proposed in the literature.

Among the first open-set recognizers proposed in the literature, it is worth mentioning the Extreme Value Machine (EVM) that models the probability distribution of the distances of each instance w.r.t. the decision margin [11]. Each class decision boundary is then represented by a subset of training data called extreme vectors. Further approaches are variants of support vector machines based on the Weibull distribution [4,13]. In [12], the authors proposed an approach called Compact Abating Probability (CAP), where the class probability of a test instance decreases according to the distance from the training instances of the class. The learning algorithm establishes a threshold on the probability value according to a CAP model based on a one-class SVM to reject far-away test instances from the training ones and delegates the final decision to a further CAP model based on the binary SVM and Extreme Value Theory.

Recently, with the growing interest in deep learning models, open-set recognizers based on neural networks have been proposed. For instance, OpenGAN has been one of the most promising models extending generative adversarial networks, which augments the training set with synthetic data representing the rest of the world, and as such not belonging to any training class, before training the model [6]. In [1], the authors use a neural network classifier to define the boundaries between known-class instances and those belonging to unknown classes by allowing the model to estimate the similarity between data and stored knowledge. Additionally, the model determines a similarity value for unknown-class instances. Further distance-based approaches based on a modified version of the nearest neighbor classifier have been proposed [5]: for a given test instance, the inference algorithm computes the distances w.r.t the two nearest neighbors whose class labels differ. The algorithm compares the ratio between these distance values against a threshold: the test instance will have the class label of the nearest neighbor only if the distance ratio exceeds the threshold.

Recent works combine the problem of open-set recognition with *prototype mining* to generate tight boundaries between the known and unknowns thanks to prototypes that determine intra-class features representations using prototypes instead of inter-class features) [9,14]. In particular, in [9], the authors focus on the problem related to the sub-optimality of the prototypes and propose an embedded-based approach to induce high-quality prototypes (according to diversity and robustness criteria) optimizing accordingly the embedding space to better discern known-class instances from those belonging to unknown classes.

Non-negative Matrix Factorization (NMF) is a paradigm for dimensionality reduction that has gained popularity due to its ability to obtain parts-based representation and enhance interpretability. NMF works by decomposing a non-negative data matrix into two non-negative reduced matrices, allowing for the discovery of patterns in high-dimensional datasets. There have been many developments and improvements to NMF algorithms in recent years.

Among the first NMF algorithms, we find those based on multiplicative approaches in which the reduced matrices are iteratively searched by multiplicative updates. In particular, the Multiplicative Updates (MU) algorithm, proposed in [7], is a well-known method that has many interesting features: it is simple to implement and can be adapted to popular variants such as sparse Non-Negative Matrix Factorization [10]. The MU algorithm has been extensively used to estimate the basis and coefficient matrices in NMF problems under a wide range of divergences and regularizers [16].

Alternative to multiplicative approaches, we found methods based on coordinate descent, such as [2], based on hierarchical alternating least squares algorithms. The coordinate descent approach to NMF involves adopting the idea of sequential coordinate-wise descent to NMF to increase the convergence rate. This approach has been shown to converge faster than well-known methods [8].

3 Basics

In this section, we separately state the NMF problem and the OSR problem, before discussing how to empower OSR approaches with NMF feature reduction capabilities. In the following, we will consider text corpora of n documents over a dictionary of m distinct terms, so that we term the associate tf-idf term-document matrix as $V \in \mathbb{R}^{n \times m}$. For simplicity, $v_{[i,:]} \in \mathbb{R}^m$ refers to the i-th row (document) vector from V.

3.1 Non-negative Matrix Factorization

Let $V \in \mathbb{R}^{n \times m}$ be a real-valued non-negative matrix, and $r \in \mathbb{N} << min(n, m)$. The r-rank NMF is the decomposition of V into the product of two non-negative matrices $W \in \mathbb{R}^{n,r}$ and $H \in \mathbb{R}^{r,m}$ such that $V = WH$.

In practice, finding the exact decomposition of V is challenging, therefore the equality constraint is relaxed into $V \simeq WH$. Furthermore, like many other matrix factorization techniques, W and H are found by solving a constrained optimization problem whose solution is found by minimizing the approximation error, i.e. the Frobenius norm, between V and WH, that is:

$$\underset{W \in \mathbb{R}^{n \times r}, H \in \mathbb{R}^{r \times m}}{\arg\min} \frac{1}{2} \|V - WH\|_F^2$$

subject to $W \geq 0$ and $H \geq 0$. Typically, the optimization problem is solved by multiplicative algorithms where W and H, initially random, are iteratively updated by minimizing the Frobenius norm. The algorithm halts when the Frobenius norm converges or, as an alternative, when reaching the maximum number of iterations.

3.2 Open-set Recognition

Let $\mathcal{U}$ be the universe of all the possible classes, $K \subseteq \mathcal{U}$ and $U \subseteq \mathcal{U} - K$ be the subsets of known and unknown classes, resp. Let $v \in \mathbb{R}^n$ be a generic example labeled as $y \in (K \cup U)$. Then, the OSR $f_{osr} : \mathbb{R}^n \longrightarrow K \cup \{\perp\}$ is a multiclass classifier in $|K| + 1$ classes, mapping v to class $f_{osr}(v) = k$ where $k \in K$ or $f(v) = \perp$, otherwise (where $\perp$ is an auxiliary label representing unknown-class membership).

Specifically, f_{osr} is a function whose analytical form is unknown in advance and requires to be approximated from training data. In particular, the foundational assumption behind OSR is that unknown-class examples never arise in training data $D_{train} = \{(v_i, y_i)\} \subseteq X \times K$ but only in inference data $D_{test} = \{(v_j, y_j)\} \subseteq X \times (K \cup U)$. Consequently, learning an OSR model means estimating f_{osr} from D_{train} without prior knowledge of unknown classes U. The resulting model is expected to correctly classify instances from D_{test}, even those belonging to unknown classes not seen in D_{train}, thus labeling them as $\perp$. OSR f_{osr} is fitted by minimizing the sum of the empirical risk and the open-space risk over D_{train}, as in the following:

$$f_{osr} = \arg\min_{f \in \mathcal{H}} R_{\mathcal{O}}(f, D_{train}) + \lambda R_{\epsilon}(f, D_{train})$$

where λ is a regularization term and $\mathcal{H}$ is the set of all possible OSRs that can fit over D_{train}. The term R_ϵ refers to the empirical risk over D_{train} (i.e., the expected risk of incorrectly classify examples in D_{train}). Then, the term $R_{\mathcal{O}}$ refers the open-space risk over D_{train} and it is defined as:

$$R_{\mathcal{O}}(f, D_{train}) = \frac{\int_{\mathcal{O}} f(x)dx}{\int_{\mathcal{S}} f(x)dx}$$

Open-space risk assumes that i) the space far from data belonging to known classes should be considered as open space $\mathcal{O}$, and ii) that labeling any instance in this space as an arbitrary known class naturally implies a risk that should be minimized. It quantifies the relative volume of training instances from the open space $\mathcal{O}$ classified as belonging to one known class $y \in K$, with respect to the instances from the closed space $\mathcal{S}$ (i.e., D_{train}) classified as known-class instances [3]: the more instances are classified as known classes in $\mathcal{O}$, the higher the open space risk. However, as instances belonging to unknown classes do not occur in training, it is often difficult to quantitatively analyze open space risk. Alternatively, the open-space $\mathcal{O}$ can be approximated by the set of points distant at most $d \in \mathbb{R}$ from any other instance in D_{train}.

3.3 Proposed approach

When used on text data, OSR algorithms recognize documents based on their associated tf-idf document vectors. Instead, our intuition is to learn the OSR using the reduced data representation resulting from the r-rank NMF on the overall term-document matrix associated with D_{train}. To do so, documents v_i in D_{train} are juxtaposed as subsequent document vectors $v_{[i,:]}$ in the term-document $V \in \mathbb{R}^{n \times m}$. Then, the matrix is factorized by computing W and H, and f_{osr} is learnt on top of W (instead of the whole matrix V). By doing so, since the NMF is computed on top of only documents from D_{train}, the OSR f_{osr} is still learnt on top of known-class documents only. However, this time, the learning procedure takes into account reduced representations of documents, that is row vectors from W whose conceptual features are linear combinations, in the basis provided by column vectors from H, of the original term vectors from V.

This way, the recognition capability of f_{osr} is augmented, in the sense that it should be able to recognize texts based on conceptual features provided by the NMF. Because of this, document vectors v arriving at inference time, and possibly belonging to unknown classes not seen during training (as those in D_{test}), should be first transformed into the corresponding reduced vectors $\hat{v}$ in terms of the conceptual features. Only after this initial pre-processing, documents are recognized by computing $f_{osr}(\hat{v})$ and the resulting label is returned.

4 Experiments

In this section, we present the results of a comparative evaluation between 2 competitor approaches: an OSR algorithm without NMF (baseline method) and an OSR algorithm equipped with an NMF algorithm. Specifically, their performance has been assessed on different publicly available datasets trying to answer the following research questions:

- How does NMF affect the OSR accuracy when varying the rank?
- How does NMF affect the OSR accuracy when increasing the number of unknown classes?

The experiments have been executed in Jupyterlab 4.0.1 environment, using Scikit-learn 1.2.2 and running Python 3.11.3, provisioned as a pod on a Kubernetes cluster. The pod was equipped with 3GB of RAM and was allowed to utilize the 10% of a single Intel Xeon Gold 5220R CPU @ 2.20GHz.

4.1 Dataset

We considered two datasets: the *Kaggle Legal Clauses* dataset and the *CUAD* dataset. Kaggle Legal Clauses is a publicly available dataset[1] concerning the

[1] `https://www.kaggle.com/datasets/mohammedalrashidan/contracts-clauses-datasets`

Dataset	LP	K	U	#instances
Legal Clauses	LP1	interest, base-salary, investment-company-act, taxes, payment, investments	ownership-of-shares, compensation, capitalization, loans, definitions, headings	11475
	LP2	whereas, entire, assignment, representations, counterparts, termination	severability, now, miscellaneous, insurance, indemnification, confidentiality	6382
CUAD	LP1	license-grant, audit-rights, anti-assignment, cap-on-liability, insurance, governing-law	revenue-profit-sharing, post-termination-services, minimum-commitment, exclusivity, rofr-rofo-rofn, ip-ownership-assignment	6974
	LP2	volume-restriction, warranty-duration, covenant-not-to-sue, uncapped-liability, parties	competitive-restriction-exception, liquidated-damages, non-transferable-license, joint-ip-ownership, no-solicit-of-employees	1488

Table 1: Learning problems over Legal Clauses and CUAD datasets

financial domain for supervised learning problems: each labeled instance is a sentence extracted from financial contracts. The CUAD dataset[2] was originally designed for solving query-answering tasks. It consists of 510 text files and a further CSV file with 510 rows and 83 columns. The CSV file associates documents, one per row (i.e., each row points to the content of a specific text file), to questions, one per column. Specifically, for each of the 42 questions, the dataset provides both i) the answer provided by human experts and ii) the list of sentences contributing to a possible answer. In the experiments, we only considered the sentences in the 41 columns not associated with answers given by human experts. This way, we annotated every sentence according to the respective column header. After removing duplicate clauses, null values, and, in the case of CUAD, also the columns with dates, and short sentences (with less than 10 words), the resulting datasets contain, respectively, i) 21187 sentences labeled considering 42 classes in the case of Legal Clauses and ii) 25709 sentences labeled according to 38 classes in the case of CUAD.

Since the NMF is too expensive to compute on a large number of instances, adopting it on the whole dataset is impractical. Therefore, we repeated the experiments on 2 different pre-defined subsets K and U, i.e. learning problems (LP), from each dataset, as reported in Table 1). We first examined the distribution of classes in each dataset. Then we created LP1 by including only the top 6 most numerous classes in K and the next 6 in U. LP2 was created using the third group of 6 most numerous classes as K and the following 6 classes as U. For each learning problem, we sampled at most 250 instances per class.

4.2 Settings

We trained the OSR using 80% of the examples and used the remaining 20% as a validation set. We fit the OSR using only observations of known classes and evaluated the model on the entire validation set (including the unknown-class instances). We considered the kNN version for open-set recognition proposed in

[2] `https://github.com/TheAtticusProject/cuad`

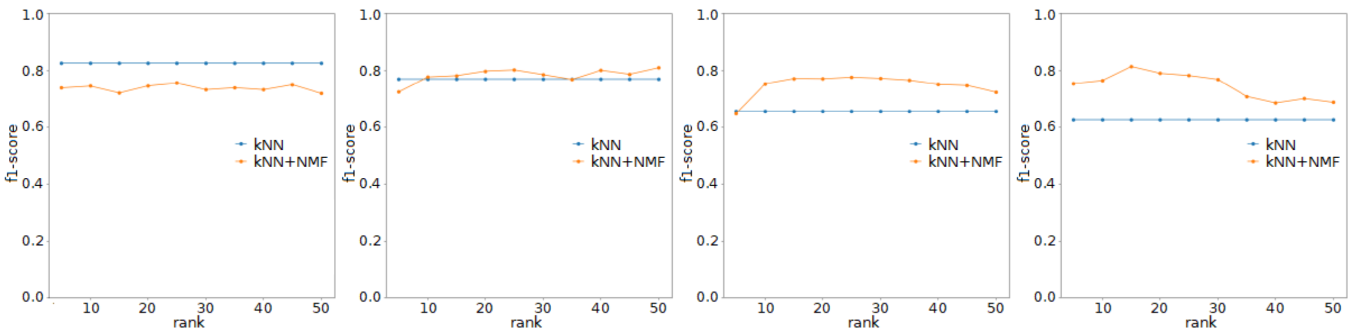

Fig. 2: F1-score of kNN and kNN+NMF over increasing NMF rank r on Legal Clauses LP1, Legal Clauses LP2, CUAD LP1 and CUAD LP2 (left to right).

[5] both with the NMF (termed as kNN+NMF hereafter) and without NMF (termed as kNN). In both cases, the algorithm required setting a threshold T to decide about the class label for the test instance: we fixed $T = 0.9$. As for kNN+NMF, the NMF has been randomly initialized (seed equal to 1) and we subsequently increased the rank r from 5 to 50 by 5 at a time.

4.3 Accuracy for increasing NMF rank

Concerning the first research question, i.e. the effect of the rank of the matrix resulting from the NMF on the accuracy, the OSR performance has been evaluated by adopting a multiclass classification macro-averaged (over the known classes) one-vs-all F1-score. Figure 2 illustrates some of the trends.

An aspect worth mentioning is that, although rank varies between 5 and 50, we have also considered larger values up to 3000. In particular, for ranks larger than 50, we observed a very rapid decreasing trend hitting an F1-score of less than 0.4. We deem these results not interesting, and therefore we have not included them in the plots, since a fair comparison should only consider ranks around optimal values of r, which in our case fall between 5 and 50. Firstly, it is straightforward to note that the F1-score of the kNN without applying NMF does not depend on the rank r and, therefore, is constant on every considered dataset. It ranges from 0.6 to more than 0.8. A more interesting case is instead observable using kNN+NMF. Indeed, we note that kNN+NMF outperforms kNN, in terms of F1 score, in 3 out of 4 cases. For these cases, the F1-score difference is maximized at rank 50, 25, 15 for Legal Clauses LP2, CUAD LP1, and LP2, respectively. In the considered rank range, we observe that the F1-score: i) has a fluctuating trend in the case of Legal clauses LP1 and LP2, ii) rapidly drops in the case of CUAD LP2 while it is almost constant in the case of CUAD LP1. Complementarily, in 1 out of 4 cases, that is Legal Clauses LP1, the optimal F1-score hit by kNN+NMF does never exceed the baseline kNN, which remains more accurate. In this case, the sampling performed when selecting documents may have affected the results: the selected instances could not be representative enough to build discriminative latent features for the known classes.

4.4 Accuracy on the increasing number of unknown classes

Concerning the second research question, we wanted to assess whether NMF-equipped OSR is more accurate than a pure OSR approach over an ever-increasing number of unknown classes. In particular, we compared the two OSR competitors trained in the previous sections, on each considered dataset, by measuring the i) F1-score, and ii) *accuracy on unknown samples* (AUS) defined as the ratio $\frac{tu}{tu+fu}$ of correctly recognized unknown-class instances (tu) to the total number of those recognized by the model ($tu + fu$) (where fu is the number of false unknown cases) on subsequent incremental variations of the validations sets. Specifically, every variation was built by controlling the associated *openness*, computed as [5]:

$$\text{openness} = 1 - \sqrt{\frac{|K|}{|K \cup U|}}$$

Openness is a complexity measure of a given dataset/learning problem, it gives an idea of how heterogeneous data can be at inference time. Unfortunately, it can only be assessed under controlled conditions due to the lack of available knowledge about unknown classes in real-world scenarios. So, assessing model accuracy w.r.t. this dimension requires a proper preparation step to obtain datasets with varying degrees of openness. To this purpose, we first removed instances of the unknown class from the dataset for each learning problem, resulting in a new dataset with an openness of 0: note that openness equal to 0, i.e. when $|U| = 0$, implies a traditional multi-class classification problem. Stemming from this dataset, we gradually built a new one adding instances of a given unknown class, one class at a time, thus resulting in 6 variations of each learning problem. The strategy is a non-exhaustive way to balance the complexity of dataset generation: multiple datasets for the same openness value with different combinations of known/unknown classes are actually possible. Indeed, exhaustive approaches to investigate the OSR accuracy w.r.t. the openness would require considering the power set of U, which can be very large.

Figure 3 and 4 reports, for each learning problem, the F1 and AUS scores for kNN and kNN+NMF one. Two expected results arise: i) consistently with previous results from the OSR literature [5], the F1-score (Figure 3) and the AUS 4 exhibit a decreasing and increasing trend over the openness, respectively, and ii) consistently with previous results from the NMF literature [15], in traditional multiclass classification setting, that is at openness 0, the F1-score of kNN+NMF is larger than kNN.

As for the F1 scores, it is evident that kNN+NMF exhibits a more rapid decreasing trend than pure kNN, despite being more accurate. In particular, in 3 out of 4 datasets, despite the F1 drop, KNN+NMF still outperforms KNN with a remarkable difference of even more than 0.20 on CUAD LP1 and LP2. On the contrary, in 1 out of 4 cases, namely in Legal Clauses LP1, we noted an

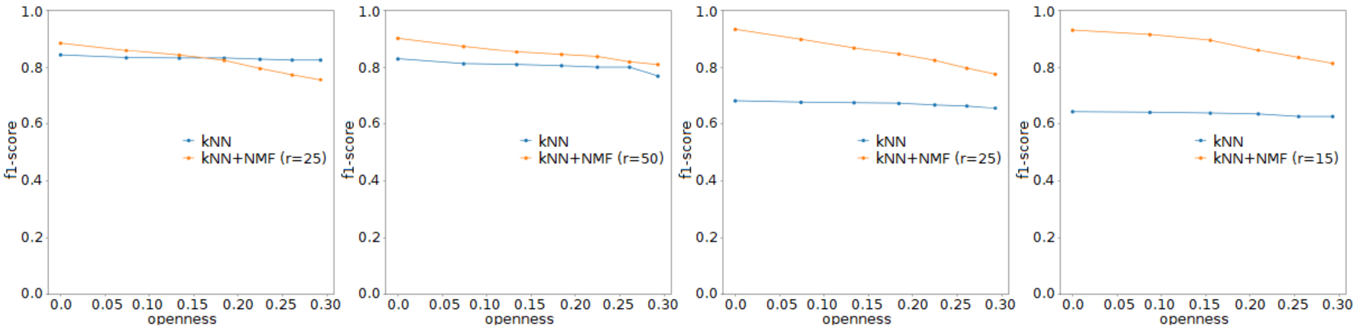

Fig. 3: F1-score of kNN and kNN+NMF over increasing openness on Legal Clauses LP1, Legal Clauses LP2, CUAD LP1 and CUAD LP2 (left to right).

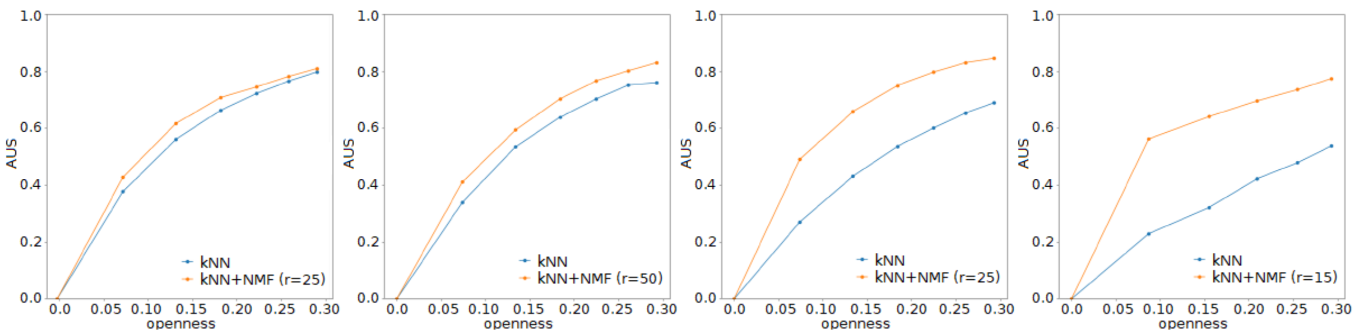

Fig. 4: AUS of kNN and kNN+NMF over increasing openness on Legal Clauses LP1, Legal Clauses LP2, CUAD LP1 and CUAD LP2 (left to right).

opposite situation: for openness values larger than 0.20, kNN+NMF does not outperform kNN. As for the AUS scores, we observe that kNN+NMF always outperforms kNN throughout the experiments, regardless of the openness. In particular, the AUS improvement is considerably larger on learning problems built from the CUAD dataset than those built from Legal Clauses. Therefore, since kNN+NMF seems to be more accurate than kNN in correctly recognizing unknown-class instances we deem kNN+NMF a favorable solution to kNN. When jointly looking at these considerations, applying NMF in the context of OSR suggests that NMF: i) may help to increase the ability to correctly recognize known-class instances but only in cases with a limited openness, ii) improves the ability to correctly recognize unknown-class instances at the price of limiting the recognition accuracy on known-class instances.

5 Conclusions

In this work, we have investigated the role of non-negative matrix factorization as a dimensionality reduction approach for improving the accuracy of the open-set recognition model on text data. Starting from previous results in the context of multiclass classification, we claimed that NMF could improve the recognition accuracy of OSR algorithms. To verify our claims, we adopted a widely known off-the-shelf NMF algorithm as a data pre-processing step and for a state-of-the-art OSR solution based on a modified version of the k-nearest neighbor algorithm. We performed two comparative evaluations: the first one aimed at determining the role of the NMF rank on the OSR accuracy over the known classes, while the second one aimed at determining the accuracy of the proposed solution involving a non-negative matrix factorization w.r.t. multiple scenarios whose openness have been gradually increased. The evaluation showed that the NMF rank affects the accuracy of OSR models on different datasets. In particular, results have shown that NMF increases the OSR accuracy towards unknown-class instances, making the approach more suitable for detecting outliers in an open-set scenario, while it rapidly decreases the accuracy towards known-class instances, making the approach only suitable within limited openness.

This research provides a preliminary insight into the role of dimensionality reduction in the context of open-set recognition and it can be extended in several ways. Firstly, we can extend the evaluation by considering further dimensionality reduction methods. Secondly, we can consider further state-of-the-art OSRs, such as EVMs, and NMF algorithms as well. Lastly, we can consider new datasets or learning problems with a larger number of known/unknown classes to gain further evidence.

Acknowledgements We acknowledge the support from the PIGRECO ("PrIvacy e Gestione REti di COllaborazione") project, co-funded by "Regolamento regionale della Puglia per gli aiuti in esenzione n. 17 del 30/09/2014 (BURP n. 139 suppl. del 06/10/2014) e s.m.i. titolo II capo 2 del regolamento generale "Aiuti ai programmi integrati promossi da medie imprese ai sensi dell'articolo 26 del Regolamento".

References

1. Cardoso, D.O., França, F., Gama, J.: A bounded neural network for open set recognition. In: 2015 International Joint Conference on Neural Networks (IJCNN). pp. 1–7 (2015). https://doi.org/10.1109/IJCNN.2015.7280680
2. Cichocki, A., Phan, A.H.: Fast local algorithms for large scale nonnegative matrix and tensor factorizations. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **92-A**(3), 708–721 (2009). https://doi.org/10.1587/transfun.E92.A.708, `https://doi.org/10.1587/transfun.E92.A.708`
3. Henrydoss, J., Cruz, S., Rudd, E.M., Gunther, M., Boult, T.E.: Incremental open set intrusion recognition using extreme value machine. In: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA). pp. 1089–1093 (2017). https://doi.org/10.1109/ICMLA.2017.000-3
4. Jain, L.P., Scheirer, W.J., Boult, T.E.: Multi-class open set recognition using probability of inclusion. In: European Conference on Computer Vision (2014)

5. Júnior, P.R.M., de Souza, R.M., de Oliveira Werneck, R., Stein, B.V., Pazinato, D.V., de Almeida, W.R., Penatti, O.A.B., da Silva Torres, R., Rocha, A.: Nearest neighbors distance ratio open-set classifier. Mach. Learn. **106**(3), 359–386 (2017). https://doi.org/10.1007/s10994-016-5610-8, https://doi.org/10.1007/s10994-016-5610-8
6. Kong, S., Ramanan, D.: Opengan: Open-set recognition via open data generation. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 793–802 (2021)
7. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Leen, T.K., Dietterich, T.G., Tresp, V. (eds.) Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA. pp. 556–562. MIT Press (2000), https://proceedings.neurips.cc/paper/2000/hash/f9d1152547c0bde01830b7e8bd60024c-Abstract.html
8. Lin, X., Boutros, P.C.: Optimization and expansion of non-negative matrix factorization. BMC Bioinform. **21**(1), 7 (2020). https://doi.org/10.1186/s12859-019-3312-5, https://doi.org/10.1186/s12859-019-3312-5
9. Lu, J., Xu, Y., Li, H., Cheng, Z., Niu, Y.: PMAL: open set recognition via robust prototype mining. In: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022. pp. 1872–1880. AAAI Press (2022), https://ojs.aaai.org/index.php/AAAI/article/view/20081
10. Pham, M., Cohen, J., Chonavel, T.: A fast multiplicative updates algorithm for non-negative matrix factorization. CoRR **abs/2303.17992** (2023). https://doi.org/10.48550/arXiv.2303.17992, https://doi.org/10.48550/arXiv.2303.17992
11. Rudd, E.M., Jain, L.P., Scheirer, W.J., Boult, T.E.: The extreme value machine. IEEE Trans. Pattern Anal. Mach. Intell. **40**(3), 762–768 (2018). https://doi.org/10.1109/TPAMI.2017.2707495, https://doi.org/10.1109/TPAMI.2017.2707495
12. Scheirer, W.J., Jain, L.P., Boult, T.E.: Probability models for open set recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(11), 2317–2324 (2014). https://doi.org/10.1109/TPAMI.2014.2321392
13. Scheirer, W.J., de Rezende Rocha, A., Sapkota, A., Boult, T.E.: Toward open set recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(7), 1757–1772 (2013). https://doi.org/10.1109/TPAMI.2012.256
14. Shu, Y., Shi, Y., Wang, Y., Huang, T., Tian, Y.: P-odn: Prototype-based open deep network for open set recognition. Scientific Reports **10** (04 2020). https://doi.org/10.1038/s41598-020-63649-6
15. Silva, C., Ribeiro, B.: Knowledge extraction with non-negative matrix factorization for text classification. In: Corchado, E., Yin, H. (eds.) Intelligent Data Engineering and Automated Learning - IDEAL 2009, 10th International Conference, Burgos, Spain, September 23-26, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5788, pp. 300–308. Springer (2009). https://doi.org/10.1007/978-3-642-04394-9_37, https://doi.org/10.1007/978-3-642-04394-9_37
16. Zhao, R., Tan, V.Y.F.: A unified convergence analysis of the multiplicative update algorithm for regularized nonnegative matrix factorization. IEEE Trans. Signal Process. **66**(1), 129–138 (2018). https://doi.org/10.1109/TSP.2017.2757914, https://doi.org/10.1109/TSP.2017.2757914