Journal of Machine Learning Research 18 (2017) 1-5 Submitted 3/17; Revised 7/17; Published 10/17

openXBOW – Introducing the Passau Open-Source Crossmodal Bag-of-Words Toolkit

Maximilian Schmitt MAXIMILIAN.SCHMITT@UNI-PASSAU.DE
Björn Schuller[*] BJOERN.SCHULLER@UNI-PASSAU.DE
Chair of Complex and Intelligent Systems, University of Passau, 94032 Passau, Germany

Editor: Geoff Holmes

Abstract

We introduce openXBOW, an open-source toolkit for the generation of bag-of-words (BoW) representations from multimodal input. In the BoW principle, word histograms were first used as features in document classification, but the idea was and can easily be adapted to, e. g., acoustic or visual descriptors, introducing a prior step of vector quantisation. The openXBOW toolkit supports arbitrary numeric input features and text input and concatenates computed sub-bags to a final bag. It provides a variety of extensions and options. To our knowledge, openXBOW is the first publicly available toolkit for the generation of crossmodal bags-of-words. The capabilities of the tool have been exemplified in different scenarios: sentiment analysis in tweets, classification of snore sounds, and time-dependent emotion recognition based on acoustic, linguistic, and visual information, where improved results over other feature representations were observed.

Keywords: bag-of-words, multimodal signal processing, histogram feature representations, feature learning

1. Introduction

The bag-of-words (BoW) principle is a common practice in *natural language processing* (NLP) (Weninger et al., 2013). In this method, *word histograms* are generated, i. e., within a text document, the frequencies of each word from a dictionary are counted. The resulting *term frequency* (TF) vector is then input to a classifier, such as a *support vector machine* (SVM), i. e., a machine learning scheme which is known to cope well with possibly irrelevant features and large, yet sparse feature vectors.

BoW has been adopted by the visual computing community, where it is known under the name bag-of-visual-words (BoVW) (Wu et al., 2011). Instead of lexical words, local image features are extracted from an image and their general distribution is modelled by a histogram. In recent years, the principle has also been employed successfully in the field of audio classification, where it is known under the term bag-of-audio-words (BoAW). Acoustic *low-level descriptors* (LLDs), such as *mel-frequency cepstral coefficients*, are extracted from the audio signal; then, the LLD vectors from single frames are quantised according to a codebook (Pancoast and Akbacak, 2014). This codebook can be the result of, e. g., a k-means clustering (Pokorny et al., 2015) or a random sampling of LLDs (Rawat et al., 2013). A histogram finally describes the distribution of the codebook vectors over the whole audio segment. Major applications of BoAW are acoustic and multimedia event detection

[*] B. Schuller is also with GLAM – Group on Language, Audio & Music, Imperial College London, U. K.

(Rawat et al., 2013; Pancoast and Akbacak, 2013; Lim et al., 2015), but they have also been successfully employed for music information retrieval (Riley et al., 2008), emotion recognition (Schmitt et al., 2016b), and medical diagnosis (Schmitt et al., 2016a).

In this contribution, we introduce the first open-source toolkit for the generation of BoW representations across modalities, thus named openXBOW ("open crossbow" – 'X' stands for CROSSmodal). The motivation behind openXBOW is to ease the generation of a fused BoW-based representation from different modalities. These modalities can be the acoustic or the visual domain, providing numeric LLDs, and written documents or transcriptions of speech, providing text or *symbolic* input. However, arbitrary modalities, such as, e. g., stemming from physiological measurement or feature streams as used in brain computing, can be processed. In case of multimodal or 'crossmodal' usage, the output of the toolkit is a concatenated feature vector consisting of histogram representations per modality or combinations of these. Crossmodal BoW have already been employed, e. g., for depression monitoring (Joshi et al., 2013), exploiting both the audio and the video domain.

openXBOW provides a multitude of options, e. g., different modes of vector quantisation, codebook generation, TF weighting, and methods known from natural language processing to process the textual features. To the knowledge of the authors, such a toolkit has not been published, so far, whereas there are already some libraries implementing BoVW, such as *DBoW2* (Gálvez-López and Tardos, 2012).

In the next section, we give an overview of the openXBOW tool, its structure and its options. In Section 3, we give results from an exemplary application of the tool. We conclude and give an outlook on future developments in openXBOW in Section 4.

2. Overview

openXBOW is implemented in Java and can thus be used on any common platform. It has been published on GitHub as a public repository[1], including both the source code and a compiled jar file for users who do not have a Java Development Kit installed. The software and the source code are published under GPLv3. openXBOW supports three formats for input and output: *CSV* (comma separated values), *ARFF* (attribute-relation file format), used in the broadly applied machine learning software *Weka* (Hall et al., 2009), and *LIBSVM* file format (only for output), used in LIBSVM (Chang and Lin, 2011) and LIBLINEAR (Fan et al., 2008). All options, which are configured through the command line call, are displayed when calling openXBOW ('java -jar openXBOW.jar').

The input is processed in the way shown in Figure 1. First, the input features can be pre-processed, particularly, (min-max)-normalisation or standardisation can be applied to the numeric descriptors, which is meaningful in case feature types with (significantly) different ranges of values are combined. The corresponding parameters are stored in the *codebook file*, in order to be applied also to a given test file (online approach).

For codebook generation, there are four different methods; *random sampling*, which means that the codebook vectors are picked randomly from the input, whereas *random sampling++* favours far-off vectors as proposed by Arthur and Vassilvitskii (2007). This implements basically the initialisation step of *k-means++ clustering*, which is also available besides the classical *k-means clustering*. The codebook generation can also be done in a *supervised* manner, learning a codebook from the input descriptors of one class separately,

[1] `https://github.com/openXBOW/openXBOW`

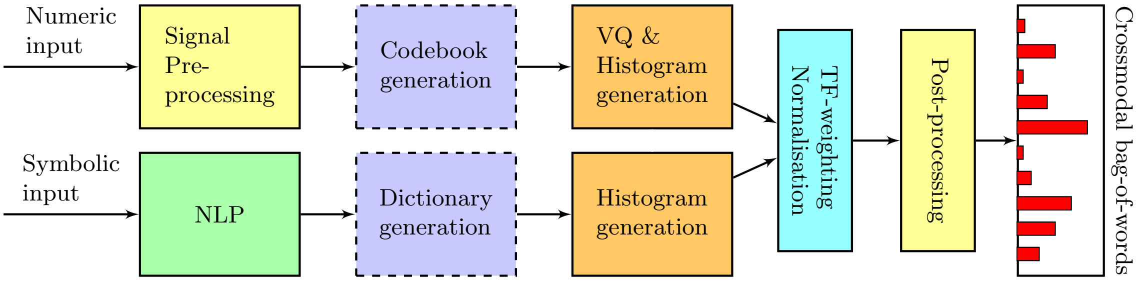

Figure 1: Overview of the basic workflow of openXBOW.

first, and then concatenating these codebooks to form a *super-codebook* (Grzeszick et al., 2015). Besides, it is also possible to split the input into well-defined sub-vectors in order to generate different sub-codebooks for different feature types. The vector quantisation (VQ) of the input is based on the minimum *Euclidean distance* between an input vector and all words in the codebook, sometimes referred to as *templates*. The bag-of-words are then generated from the term frequencies, i. e., the number of occurrences of each template, within the whole document. In the input file, all feature vectors belonging to the same instance need to have the same unique name as an attribute. The input can also be subject to *multiple assignments*, where not only the word with the closest distance is considered but a specified number of close templates. A way of doing *soft* vector quantisation is applying *Gaussian encoding* (Pancoast and Akbacak, 2014), where in each word assignment step, the increment is weighted with the distance to the templates.

For the text domain, standard processing techniques known from NLP are available, such as *stopping*, *n-grams*, and *n-character-grams*. The numeric codebook and the symbolic dictionary are stored in a file to be used for independent test data.

The BoW representations of different domains are fused into a single feature vector (early fusion). Logarithmic TF-weighting and inverse document-frequency (IDF) weighting can be applied (Riley et al., 2008). The bags can then be normalised, a step which might be meaningful if the input instances have different lengths. Finally, post-processing in terms of (min-max-)normalisation and standardisation is available for the resulting BoW.

3. Experiments

The usage of openXBOW is now shortly exemplified in one scenario: *crossmodal emotion recognition*. The above mentioned GitHub repository includes a *tutorial* which allows the user to reproduce results for different scenarios and several features types, including *sentiment analysis* of *tweets* in a data set of more than 1 million instances.

Emotion recognition in speech has been conducted on the SEWA[2] (Automatic Sentiment Analysis in the Wild) corpus, more specifically, on video chat recordings of 64 German subjects. The overall length of this audio-visual data is approximately 89 minutes, the data was split into subject-independent training, development (devel), and test partitions (34/14/16 subjects). Emotion has been annotated continuously in terms of the two emotional dimensions *arousal* and *valence* (Schmitt et al., 2016b). Results are presented in terms of the *concordance correlation coefficient* (CCC) between the prediction and the annotation. For

[2] `http://sewaproject.eu/`

Modality	Arousal (devel)	Arousal (test)	Valence (devel)	Valence (test)
Acoustic	.535	.470	.402	.426
Visual	.434	.314	.385	.344
Linguistic	.364	.293	.328	.320
Crossmodal (early fusion)	.560	.432	.536	.509
Crossmodal (late fusion)	.588	.499	.495	.523

Table 1: Performance (in terms of CCC) of emotion recognition on the SEWA German video chats using crossmodal BoW. The codebook size for numeric features is 1 000, the number of assignments is 10. The dictionary consists of 346 words.

the acoustic domain, the 65 LLDs from the ComParE feature set have been extracted using the toolkit openSMILE (Eyben et al., 2013). For the visual domain, we used 49 *facial landmarks* extracted with the Chehra Face Tracker (Asthana et al., 2014). Manual transcriptions of the speech were used as linguistic input. As the target of the prediction is time-dependent, the input was segmented into overlapping blocks of 8 *s* width and 0.1 *s* hop size, as described by Schmitt et al. (2016b). For decoding of the BoW, LIBLINEAR was used; the complexity parameter was optimised on the development set. The results reported in Table 1 show that, both early and late fusion—by training another SVM based on the predictions from single modalities—are suitable for the prediction of emotional speaker states. Further results with BoAW on well-established databases with comparison to other approaches have been reported by Schmitt et al. (2016a,b).

The performance of openXBOW is quite decent. On a system with an *Intel Core i7-4770 (3.4 GHz)* CPU, *16 GB RAM*, *Windows 10* operating system, and *Java Version 8, Update 121*, the computation of the aforementioned crossmodal BoW (with early fusion) took 263 *s* for training and 67 *s* for prediction. Calling openXBOW for a short video segment of 8 *s* duration takes 0.57 *s*. Thus, the toolkit is suitable for real-time applications ($<$ 10 % real-time factor for crossmodal input). For the sentiment analysis task mentioned in the *tutorial*, the generation of the BoW representation for 1 million tweets took 118 *s*.

4. Conclusions and Outlook

We introduced our novel openXBOW toolkit—a first of its kind—for the generation of BoW representations from crossmodal symbolic (including text), but also numeric (such as audio or video feature streams) information representations. We showed the potential of the toolkit and the underlying BoW principle in a crossmodal emotion recognition task on a state-of-the-art database.

Future work on openXBOW will include further codebook generation techniques such as *EM clustering* or *non-negative matrix factorisation* and alternative methods of soft vector quantisation.

Acknowledgments

The research leading to these results has received funding from the European Union's Horizon 2020 and 7th Framework Programmes under GA No. 645094 (H2020 IA SEWA) and No. 338164 (ERC StG iHEARu).

References

D. Arthur and S. Vassilvitskii. K-means++: the advantages of careful seeding. In *ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, 2007.

A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Incremental face alignment in the wild. In *CVPR*, pages 1859–1866, 2014.

C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. *ACM TIST*, 2:27:1–27:27, 2011.

F. Eyben, F. Weninger, F. Groß, and B. Schuller. Recent developments in openSMILE, the munich open-source multimedia feature extractor. In *ACM MM*, pages 835–838, 2013.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: a library for large linear classification. *JMLR*, 9:1871–1874, 2008.

D. Gálvez-López and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE T-RO*, 28(5):1188–1197, 2012.

R. Grzeszick, A. Plinge, and G. A. Fink. Temporal acoustic words for online acoustic event detection. In *GCPR*, pages 142–153, 2015.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *ACM SIGKDD Explor. Newsletter*, 11(1):10–18, 2009.

J. Joshi, R. Goecke, S. Alghowinem, A. Dhall, M. Wagner, J. Epps, G. Parker, and M. Breakspear. Multimodal assistive technologies for depression diagnosis and monitoring. *Journal on MultiModal User Interfaces*, 7(3):217–228, 2013.

H. Lim, M. J. Kim, and H. Kim. Robust sound event classification using LBP-HOG based bag-of-audio-words feature representation. In *INTERSPEECH*, pages 3325–3329, 2015.

S. Pancoast and M. Akbacak. N-gram extension for bag-of-audio-words. In *ICASSP*, pages 778–782, 2013.

S. Pancoast and M. Akbacak. Softening quantization in bag-of-audio-words. In *ICASSP*, pages 1370–1374, 2014.

F. Pokorny, F. Graf, F. Pernkopf, and B. Schuller. Detection of negative emotions in speech signals using bags-of-audio-words. In *WASA/ACII*, pages 879–884, 2015.

S. Rawat, P. F. Schulam, S. Burger, D. Ding, Y. Wang, and F. Metze. Robust audio-codebooks for large-scale event detection in consumer videos. In *INTERSPEECH*, pages 2929–2933, 2013.

M. Riley, E. Heinen, and J. Ghosh. A text retrieval approach to content-based audio hashing. In *ISMIR*, pages 295–300, 2008.

M. Schmitt, C. Janott, V. Pandit, K. Qian, C. Heiser, W. Hemmert, and B. Schuller. A bag-of-audio-words approach for snore sounds' excitation localisation. In *ITG Speech Communication*, pages 230–234, 2016a.

M. Schmitt, F. Ringeval, and B. Schuller. At the border of acoustics and linguistics: bag-of-audio-words for the recognition of emotions in speech. In *INTERSPEECH*, pages 495–499, 2016b.

F. Weninger, P. Staudt, and B. Schuller. Words that fascinate the listener: predicting affective ratings of on-line lectures. *International Journal of Distance Education Technologies, Special Issue on Emotional Intelligence for Online Learning*, 11(2):110–123, 2013.

J. Wu, W.-C. Tan, and J. M. Rehg. Efficient and effective visual codebook generation using additive kernels. *JMLR*, 12:3097–3118, 2011.