Adaptive Non-negative Matrix Factorization in a Computational Model of Language Acquisition

Joris Driesen[1], *Louis ten Bosch*[2], *Hugo Van hamme*[1]

[1]Dept. ESAT, K.U.Leuven, Leuven, Belgium
[2]CLST, Radboud University, Nijmegen, The Netherlands
joris.driesen@esat.kuleuven.be, louis.tenbosch@let.ru.nl, hugo.vanhamme@esat.kuleuven.be

Abstract

During the early stages of language acquisition, young infants face the task of learning a basic vocabulary without the aid of prior linguistic knowledge. It is believed the long term episodic memory plays an important role in this process. Experiments have shown that infants retain large amounts of very detailed episodic information about the speech they perceive (e.g. [1]). This weakly justifies the fact that some algorithms attempting to model the process of vocabulary acquisition computationally process large amounts of speech data in batch. Non-negative Matrix Factorization (NMF), a technique that is particularly successful in data mining but can also be applied to vocabulary acquisition (e.g. [2]), is such an algorithm. In this paper, we will integrate an adaptive variant of NMF into a computational framework for vocabulary acquisition, foregoing the need for long term storage of speech inputs, and experimentally show its accuracy matches that of the original batch algorithm.
Index Terms: Computational models, Machine Learning, Language Acquisition, Developmental Psychology.

1. Introduction

The process of language acquisition by young infants is ultimately based on the detection and learning of word-like units from the speech signal. The discovery of these word-like units is based on the search for recurrent word-like segments in the speech signal, such that representations are hypothesized and updated when more stimuli are processed. Young infants learn that auditory stimuli such as segments of speech are not arbitrary sounds, but instead are patterns that can be associated with concepts (e.g. with objects and events in the environment). Normally this development process results in representations of what linguists call 'words'. This word discovery process is particularly interesting since infants start without any lexical knowledge and the speech signal does not contain clear acoustic cues for boundaries between words. The conventional interpretation is that infants must 'crack' the speech code ([3]) and that the discovery of word-like entities is the first step towards more complex linguistic analyses ([4]).

Several computational models of the language acquisition process have been proposed (e.g. [5]). Most of them use multimodal stimuli as input and search for repeating word-like units by cross-modal association. They assume that the behaviour of infants can be adequately described by a search for word-concept associations that statistically 'stand out' across situations. This process is also referred to as 'grounding'. Another recently proposed technique that adheres to these principles, is Non-negative Matrix Factorization (NMF) ([2], [6]). Due to its computationally and conceptually attractive properties, it has been successfully applied in machine learning and data mining. In the NMF approach, the discovery of structure is mathematically represented as a decomposition of rich high-dimensional inputs into a low-dimensional parts-based representation. In the context of modelling language acquisition, the NMF approach is interesting because it is able to find structure without any *a priori* given top-down information. It is able to find statistically relevant parts of the speech signal, based on regularities and irregularities in sound sequences (see [2]), similar to what real babies do. A few months old, infants can discriminate details such as differences between vowels and consonants (e.g. [7]). At an age of about 7 months infants can perform tasks that are similar to word segmentation (e.g. [8],[9]). However, when viewed from a cognitive perspective, the use of NMF in its original form (see [10]) in the context of language acquisition is not indisputable. It requires large numbers of speech utterances to be kept in memory, in order to converge towards appropriate matrix decompositions. This way of data presentation conflicts with the idea of language acquisition being a process of online reinforcement learning, in which new input is only observed once in order to initialize or update internal representations of speech.

In this paper, we discuss a computational model of the language acquisition process in an *adaptive* way, where the focus lies on the *gradual* emergence of word-like units. In section 2, we will briefly review the NMF-algorithm and elaborate on the necessary extensions for adaptivity. In section 3, we will describe and discuss the results of an experiment that compares the accuracy of adaptively trained word models with word models trained in batch. Finally, a discussion follows in section 4.

2. Structure discovery by NMF

NMF is a member of the family of machine learning approaches in which the discovery of structure is based on matrix decomposition. It is a factorization algorithm that decomposes a (typically large) data matrix V, of size $M \times N$, into the product of two (much smaller) matrices W and H, such that

$$V \approx W \cdot H \tag{1}$$

This approach differs from other matrix factorizations such as Principal Component Analysis or eigenvalue decomposition: V, W and H are constrained to only contain *non-negative* elements. Because of this, the NMF-decomposition can be physically and conceptually interpreted. The columns of W are the non-negative parts that can best approximate the columns of V by a weighted addition. In our context the columns of V are speech utterances, that are composed of words, represented in the columns of W. The factorization can be solved by minimiz-

ing a metric used to assess the difference between the *observed* matrix V and its *hypothesized reconstruction* WH. In this paper, we applied the Kullback-Leibler divergence between V and WH:

$$D_{KL}(V\|WH) = \sum_{i,j}\left(V_{ij}\log\frac{V_{ij}}{(WH)_{ij}} + (WH)_{ij} - V_{ij}\right) \tag{2}$$

which can be minimized by iteratively applying the multiplicative updates (see [10]):

for N iterations do

- $W_{ik} \rightarrow W_{ik}\cdot\sum_j H_{kj}\left(\frac{V}{WH}\right)_{ij}$ (3a)
- Normalize the columns of W (3b)
- $H_{kj} \rightarrow H_{kj}\cdot\sum_i W_{ik}\left(\frac{V}{WH}\right)_{ij}$ (3c)

It can be shown that this converges towards a solution with maximum likelihood $L(V|W,H)$ ([11], [12]).

2.1. Adaptive NMF

If a NMF-based decomposition technique is to be applied in a computational model of word discovery that is conceptually close to a real-life learning situation, where speech input is presented one utterance at a time, online adaptivity of that technique becomes a very interesting feature. It allows for small incremental updates towards an optimal solution without resorting to storing and processing large amounts of past input data. To achieve this adaptivity, Bayesian updating can be used in a similar way as in [13]. Suppose that there is a sequence of epochs, in each of which an input matrix $V^{(n)}$ is presented to the algorithm, consisting of a small number of columns from V. The columns of W, which are assumed to contain Dirichlet distributed probabilities, can then be reestimated in each epoch based on this input. Concretely, NMF in epoch n can be solved as a maximum likelihood problem:

$$(W^{(n)}, H^{(n)}) = \arg\max_{W,H^{(n)}} \log\left(L(V^{(n)}|W,H^{(n)})\right) + \gamma\log\left(g(W|\kappa^{(n-1)})\right) \tag{4}$$

in which $L()$ is the likelihood. $g(W|\kappa^{(n-1)})$ is the prior distribution of W, which is assumed to be Dirichlet with $(\kappa^{(n-1)}+1)$ a matrix containing its hyperparameters, estimated on the inputs from the $(n-1)$ previous epochs. Note that we have assumed $H^{(n)}$ to be independent from past inputs. This is done because in the context of this paper, the columns of W are envisaged as estimations of the word models, which are the same in every epoch, unlike the columns of H, which will estimate word activations that change along with the input data $V^{(n)}$. The parameter $0 \ll \gamma < 1$ is introduced to assign a slightly smaller weight to the past information than to the current information.

Applying this Bayesian updating principle in practice yields the following updates in each epoch n:

for N iterations do

- $W^{(n)}_{ik} \rightarrow W^{(n)}_{ik}\cdot\sum_j H^{(n)}_{kj}\left(\frac{V}{WH}\right)^{(n)}_{ij} + \gamma\cdot\kappa^{(n-1)}_{ik}$ (5a)
- Normalize the columns of $W^{(n)}$ (5b)
- $H^{(n)}_{kj} \rightarrow H^{(n)}_{kj}\cdot\sum_i W^{(n)}_{ik}\left(\frac{V}{WH}\right)^{(n)}_{ij}$ (5c)

The matrix $W^{(n)}$ is initialized with the values of $W^{(n-1)}$ at the end of the previous epoch (after N iterative updates). The matrix $H^{(n)}$ is initialized with random values. At the end of each epoch, κ is updated as follows:

$$\kappa^{(n)}_{ik} = W^{(n)}_{ik}\sum_j H^{(n)}_{kj}\left(\frac{V}{WH}\right)^{(n)}_{ij} + \gamma\cdot\kappa^{(n-1)}_{ik} \tag{6}$$

where $\kappa^{(0)}$ is initialized as a matrix containing all ones. Note that the input in a certain epoch cannot influence W indefinitely into the future. The inclusion of γ in equation 6 ensures that this influence will exponentially decay over time. Also, as a side effect, elements of κ can not increase without bounds.

3. Experiments

3.1. Experimental Setup

For our experiments we made use of a database that was recorded for the ACORNS-project[1]. It consists of 4000 English sentences with a simple syntactic structure, spoken by two male and two female speakers. Each of these utterances contains a single keyword, chosen from the following set: 'Angus', 'Ewan', 'bath', 'book', 'bottle', 'car', 'daddy', 'mummy', 'nappy', 'shoe' and 'telephone' (example: "Look at daddy"). Environmental information from other modalities than speech that accompanies each utterance is represented in a simplified way as a tag that corresponds with the keyword in that utterance. The model does not know anything about lexicon, morphology, phonetic realisations of words and of carrier phrases, but must learn to detect and update internal acoustic representations of word-like units.

3000 utterances were used in the train set, 1000 in the test set. Equal amounts of data from all four speakers were present in both sets. Each utterance is converted to a sequence of static, Δ- and $\Delta\Delta$-labels, by means of vector quantization. The codebook sizes were 150, 150 and 100 respectively. Histograms of VQ-label cooccurrences at time lags of 20ms, 50ms and 90ms were combined into high-dimensional vectors of fixed size, that were placed in the columns of the data-matrix V. This is the HAC-representation of the data (Histogram of Acoustic Cooccurrences), which is more extensively described in [2]. Since the HAC-representation *accumulates* cooccurrence counts, the cooccurrence statistics of words (columns of W) contribute additively to the cooccurrence statistics of utterances (columns of V), therefore $V \approx WH$. A condition is that W has enough columns to accomodate all keywords and some extra columns to model everything unrelated to any of them (e.g. the words in the carrier sentences). The W-matrix in our experiments had 20 columns (twice the number of keywords).

[1] http://www.acorns-project.org

Utterance level (grounding) information (i.e. the tag that represents the keyword) is added to the original V-matrix, which we will henceforth call V_{HAC}, in the form of a word identity matrix V_g. The row dimension of V_g is equal to the number of different possible keywords and its elements are

$$V_g(i,j) = \begin{cases} 1 & \text{if utt. j contains keyword i} \\ 0 & \text{otherwise} \end{cases}$$

The NMF decomposition in training then looks like:

$$\left[\begin{array}{c} V_g \\ \hline V_{HAC} \end{array}\right] \approx \left[\begin{array}{c|c} W_g & G_g \\ \hline W_{HAC} & G_{HAC} \end{array}\right] \cdot \left[\begin{array}{c} H_w \\ \hline G_H \end{array}\right] \quad (7)$$

in which W_g is initialized as a matrix with large elements on its diagonal and very small elements elsewhere. H_w is equal to V_g and W_{HAC} is initialized with random numbers, as are the garbage matrices G_g, G_{HAC} and G_H.

If we denote the data-matrix containing the utterances of the test set $V_{HAC}^{(test)}$, and the trained W-matrix obtained from equation 3a (or in the adaptive case equation 5a) $W^{(train)}$, the accuracy of this trained W-matrix can be tested in batch as follows: we update a randomly initialized matrix $H^{(test)}$ until convergence (30 iterations), applying equation 3c such that

$$V_{HAC}^{(test)} \approx \left[\begin{array}{c|c} W_{HAC}^{(train)} & G_{HAC}^{(train)} \end{array}\right] \cdot H^{(test)} \quad (8)$$

The activations of the keywords by the utterances of the test set can then be calculated as

$$A = W_g^{(train)} \cdot H^{(test)} \quad (9)$$

Since there is only one keyword to be detected in each utterance, it suffices to find the maximal element in each column of A, to determine a keyword error rate (KER).

3.2. Results with batch NMF

Training is performed until convergence according to formulas 3a, 3c and 7. In practice, this is reached after 200 iterations. To accommodate for the random elements in W and H at initialization time, which can affect the outcome of the decomposition, the entire training was performed five times. Of these five different decompositions, only the one with the lowest $D_{KL}(V\|WH)$ was used for testing. Applying the testing method from section 3.1, using equations 8 and 9, yielded a KER of 0.3%.

3.3. Results with adaptive NMF

In this experiment, not only do we use the same V-matrices during training and testing as in section 3.2, but we also use the same random initializations of W and H that was eventually selected there. In each epoch n a number of columns from V, $V^{(n)}$, is presented to the algorithm in order to update the W-matrix:

$$V^{(n)} \approx W^{(n)} \cdot H^{(n)}$$

where $H^{(n)}$ contains the corresponding columns of the initial H-matrix. In our experiments $V^{(n)}$ and $H^{(n)}$ only contained a single column from V and H respectively (representing a single utterance from the train set). 10 Iterations per epoch were performed as a trade-off between computational complexity and adequate convergence of the algorithm. Once every 20 epochs, we used the W-matrix to determine the KER on the test set, as described in section 3.1. The experiment was repeated for γ=1 (no forgetting), γ=0.999, γ=0.99 and γ=0.9. The performance with batch processing on the same amount of training data was also determined. The results are shown in figure 1. Note that the KER of *batch* NMF on low amounts of data is slightly above that of the adaptive NMF. A possible explanation is that the batch algorithm is more troubled by local optima than the adaptive one.

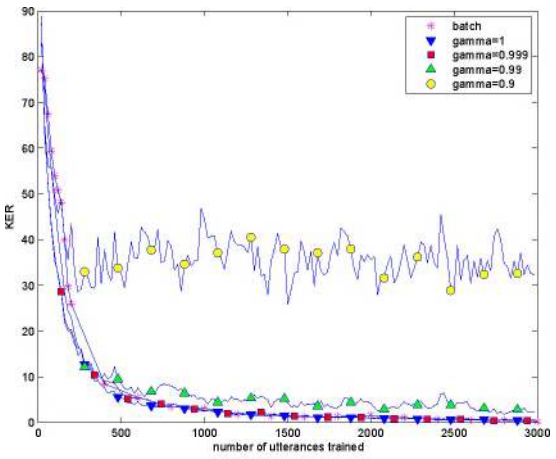

Figure 1: *The evolution of KER on a test set for the models with different γ-values, when training is done one utterance at a time. The KER obtained with batch training is also indicated.*

In a second experiment, we decided to pass through the training data in the same way as before, training on one utterance at a time and testing after every 20 utterances. However, after processing all training data once, the grounding vector was flipped (matrices V_g and H_w in equation 7) upside down after which the training data was presented a second time. This forced the algorithm to adapt the models in the W-matrix to these changes in the input. For instance, the word model for "telephone" had to be adjusted to model the word "Angus", the model for "nappy" had to change into "shoe" etc... The results are shown in figure 2.

3.3.1. About the influence of γ

As can be seen in figure 1, the KER on the test set steadily decreases towards an average level, that is dependent of γ. This is not surprising, since γ essentially determines the memory's longevity, which is proportional to $\frac{1}{1-\gamma}$. The shorter the memory, the less previous data the models are actually trained on, thus the higher the average KER. It is also clear this difference only becomes visible when the number of previous utterances (epochs) exceeds this memory length. The similarity of the algorithm's performance in this experiment when $\gamma = 0.999$ as opposed to when $\gamma = 1$ can be explained by the fact that the memory length is in the same order of magnitude as the length of the database.

Not only does memory effect overall performance, it also influences the speed with which the word models can adapt to changes in the input. The shorter the memory, the faster models are expected to adapt. This can clearly be observed in figure 2. From this figure, we can also determine the memory length to be approximately around $\frac{1}{1-\gamma}$ utterances. At that point, there is a tipping point after which the models become more sensitive

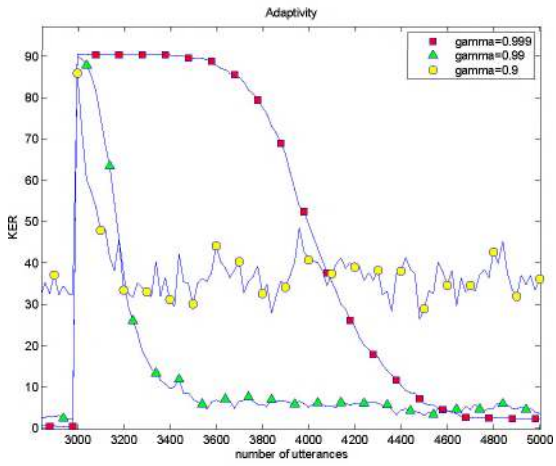

Figure 2: *The KER on the test set during incremental training of W. After 3000 utterances, the algorithm is forced to radically change its internal models*

to new inputs than to the old ones.

4. Discussion

It is very difficult to make strong claims about the long term memory of infants. The general consensus is that babies retain very distinct detailed memories of perceived speech (see [1]). It is therefore possible that at any given time during the first stages of acquiring language skills, a large number of such episodic memories is profusely processed upon by the baby's brain to hypothesize potential word-like representations. This assumption supports the cognitive plausibility of computational models that require *batch* processing of speech data. In the adaptive experiments above, however, we carefully avoided making this assumption. At any point in time, one and only one utterance is processed to refine the word representations stored in long term memory (i.e. the W-matrix), abolishing the need to store large amounts of speech. This does not imply that the detailed memory in real infants serves no purpose at all in the process of language acquisition. In fact, since the adaptive NMF-algorithm can be seen as a state machine, changing its internal state based on an input and its previous state, the presentation of the same speech utterance at different times can have very different (beneficial) effects on the word models in W. This means that the reiteration of past speech utterances stored in memory may be crucial for success in more complex learning environments. In our experiments, however, this proved to be unnecessary, since the adaptive algorithm was capable of producing word models as accurate as those obtained by batch training on the same data, without taking any training utterance as input more than once.

Another property of the adaptive NMF-algorithm that is tractable from a cognitive point of view, is that it provides an elegant model for forgetting. The experimental results show that due to this forgetting property, the algorithm can even adapt to extreme changes in the input data and that recent training data dominates over data that was presented more than $\frac{1}{1-\gamma}$ epochs ago. Analogies can be drawn with real infants adapting to e.g. speaker-specific pronunciations of words.

5. Conclusion

In this paper, we introduced a computational framework for acquiring a vocabulary in an adaptive way. The utilized algorithm was NMF, extended with the idea of Bayesian updating, proposed for PLSA in [13]. Some improvements were proposed to allow a better emulation of certain properties of the human brain, such as forgetting. Furthermore, we have experimentally shown that the word models trained by this adaptive algorithm are comparable to models trained in batch, in terms of accuracy. This means a computational model of word acquisition based on NMF does not need to rely on long term episodic memory.

6. Acknowledgments

This research is part of the ACORNS project, funded by the European Commission under contract number FP6-034362.

7. References

[1] R. Newman, "The level of detail in infant's word learning," *Current Directions in Psychological Science*, vol. 17, no. 3, pp. 229–232, 2008.

[2] H. Van hamme, "HAC-models: a novel approach to continuous speech recognition," in *Proc. Interspeech*, (Brisbane, Australia), pp. 2554–2557, september 2008.

[3] P. K. Kuhl, "Early language acquisition: cracking the speech code," *Nature Reviews Neuroscience*, vol. 5, pp. 831–843, November 2004.

[4] J. R. Saffran and D. P. Wilson, "From sybllables to syntax: Multilevel statistical learning by 12-month-old infants," *Infancy*, vol. 4, pp. 273–284, May 2003.

[5] D. Roy and A. Pentland, "Learning words from natural audio-visual input," in *In International Conference of Spoken Language Processing*, p. 1279, 1998.

[6] L. Ten Bosch, H. Van hamme, and L. Boves, "A computational model of language acquisition: focus on word discovery," in *Proc. Interspeech*, pp. 2570–2573, september 2008.

[7] P. W. Jusczyk, "How infants begin to extract words from speech," *Trends in Cognitive Sciences*, vol. 3, pp. 323–328, September 1999.

[8] J. F. Werker and H. H. Yeung, "Infant speech perception bootstraps word learning," *Trends in Cognitive Science*, vol. 9, no. 11, pp. 519–527, 2005.

[9] E. L. Newport, "Statistical language learning in human infants and adults," in *Proc. Interspeech*, (Pittsburgh, USA), September 2006.

[10] D. Lee and H. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, no. 401, pp. 788–791, 1999.

[11] E. Gaussier and C. Goutte, "Relation between plsa and nmf and implications," in *Proc. SIGIR*, (Salvador, Brazil), 2005.

[12] T. Hoffman, "Probabilistic latent semantic analysis," in *UAI*, (Stockholm), 1999.

[13] J.-T. Chien and M.-S. Wu, "Adaptive bayesian latent semantic analysis," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, pp. 198–207, january 2008.