PARTIALLY MUTUAL EXCLUSIVE SOFTMAX FOR POSITIVE AND UNLABELED DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

In recent years softmax, together with its fast approximations, has become the de-facto loss function for deep neural networks with multiclass predictions. However, softmax is used in many problems that do not fully fit the multiclass framework and where the softmax assumption of mutually exclusive outcomes can lead to biased results. This is often the case for applications such as language modeling, next-event prediction, and matrix factorization, where many of the potential outcomes are not mutually exclusive, but are more likely to be conditionally independent given the state. To this end, for the set of problems with positive and unlabeled data, we propose *Partially Mutual Exclusive Softmax (PMES)*, a relaxation of the original softmax formulation, where, given the observed state, each of the outcomes are conditionally independent but share a common set of negatives. Since we operate in a regime where explicit negatives are missing, we create a cooperatively-trained model of negatives, and derive a new negative sampling and weighting scheme which we call *Cooperative Importance Sampling (CIS)*. We show empirically the advantages of our newly introduced negative sampling scheme by plugging it into the Word2Vec algorithm and evaluating it extensively against other negative sampling schemes on both language modeling and matrix factorization tasks, where we show large lifts in performance.

1 INTRODUCTION

Learning from positive and unlabeled data is a well-defined task in machine learning, known as positive-unlabeled (PU) learning (Elkan & Noto (2008); Li & Liu (2003); Lee & Liu (2003); Ren et al. (2014); Paquet & Koenigstein (2013)). The applications of PU learning are numerous, as in many fields negative data is either too expensive to obtain or too hard to define. This is the case in language modeling, where one only observes examples of valid sentences and documents, and tries to learn a generative process. Similarly, in computer vision, with the recent work on Generative Adversarial Networks (GANs), one tries to learn the underlying generative process that produces meaningful images. In both cases, the space of negatives (e.g., non-sentences or non-images) is not clear. Similarly, in matrix factorization, most of the matrices contain pairs of observed interactions, and there are no available explicit negatives.

In all of the applications enumerated above, it is quite common to encounter solutions that are based on a softmax loss function. In language modeling, and more recently in matrix factorization, the *Word2Vec* algorithm (see Mikolov et al. (2013a)) models the conditional probability of observing all possible items in the vicinity of a context item as a categorical distribution using the softmax loss.

However, modeling the probability of co-occurrence as a categorical distribution is a very biased assumption that is clearly refuted by the data, since for all context words there are multiple words that co-occur at the same time.

In our paper we propose *Partially Mutual Exclusive Softmax (PMES)*, a new model that relaxes the mutual exclusivity constraint over the outcomes. PMES relaxes this constraint by splitting the set of all outcomes into a set of possible outcomes that are conditionally independent given the context, and a set of impossible outcomes which become negative examples.

The context-dependent negative set is hypothesized but not known, so in our method we introduce a model for negatives that is used to weight the sampled candidate negatives. The training algorithm is based on the simultaneous training of two neural networks. The first network is a generator that fits the positives and the sampled negatives. The second network is the discriminator, which is trained to separate the true positive pairs from generated pairs, and is used as the model of the probability that an example would receive a negative label.

The resulting solution has many similarities with other recent negative sampling methods for approximating full softmax. However, unlike most of the previous methods, our method is not trying to faithfully approximate the full softmax formulation, but to fix some of its over-simplifying assumptions. Furthermore, we believe that the observed lift in performance of some of the negative sampling work over the full softmax can be explained through the prism of *Partially Mutual Exclusive Softmax*.

Our hypothesis is further confirmed by experiments on language modeling and matrix factorization, where we show a big lift in performance over previous work on negative sampling and full softmax. We validate some of our intuitions on the advantages of our sampling procedure on an artificial dataset where the support sets are known, and show that our sampling scheme correctly approximates the support, which is not the case for other softmax variants.

Overall, the main contributions of this paper are the following:

- We propose *Partially Mutual Exclusive (PME) Softmax*, a modified version of the softmax loss that is a better fit for problems with positive and unlabeled data.
- We derive a new negative sampling scheme based on an cooperatively-trained models of negatives which we denote as *Cooperative Importance Sampling (CIS)*
- We show empirically the validity of our proposed approach by plugging our new loss into the Word2Vec model, and evaluating this enhanced Word2Vec model against classical sampling schemes for Word2Vec on language modeling and matrix factorization tasks across six real-world datasets.

We discuss related work on Word2Vec, negative sampling schemes for softmax and GANs in Section 2 of this paper. In Section 3 we formally introduce our *PME-Softmax* loss and the associated CIS negative sampling scheme, and describe the training algorithm. We highlight the performance of our method in Section 4, and conclude with ideas and directions for future work in Section 5.

2 Related Work

2.1 Word2Vec, softmax and negative sampling schemes

With the introduction of Word2Vec, Mikolov et al. (2013b;a) achieved state-of-the-art results in terms of the quality of the words embeddings for various language modeling tasks. In the last couple of years, Word2Vec became a widely embedding method, that is used in various setups from language modeling to matrix factorization. Since then, it is important to note the progress made in improving words embeddings (Bojanowski et al. (2016); Pennington et al. (2014)), especially more recently with the FastText model (Joulin et al. (2016)) that also leverages n-gram features.

When dealing with the training of multi-class models with thousands or millions of output classes, one can replace softmax by candidate sampling algorithms. Different methods can speed up the training by considering a small randomly-chosen subset of candidates for each batch of training examples. Gutmann & Hyvärinen (2010) introduced Noise Contrastive Estimation (NCE) as an unbiased estimator of the softmax loss, and it has been shown to be efficient for learning word embeddings Mnih & Teh (2012). In Mikolov et al. (2013b), the authors propose a *negative sampling* loss not to approximate softmax and but to learn high-quality vectors. Defined in Bengio et al. (2003), *sampled softmax* can also be used to train these language models. Benchmarked in ((Bengio & Senécal, 2008; Jean et al., 2014)), the authors have have shown state-of-the-art results in terms of performance and time complexity. Indeed, sampled softmax avoids computing scores for every possible continuation. Here, one chooses a proposal distribution from which it is cheap to sample, and performs a biased importance sampling scheme for approximating the softmax.

Most previous work has focused on generic sampling methods such as uniform sampling and unigram sampling. More recently, in Chen et al. (2018), the authors provide an insightful analysis of negative sampling. They show that negative samples that have high inner product scores with the context word are more informative in terms of gradients on the loss function. They show analytically and empirically the benefits of sampling from the unigram distribution. Leveraging this analysis, the authors propose a dynamic sampling method based on inner-product rankings.

2.2 GANs

First proposed by in 2014 in Goodfellow et al. (2014), GANs have been quite successful at generating realistic images (Ledig et al. (2016); Radford et al. (2015); Miyato et al. (2018)). GANs can be viewed as a framework for training generative models by posing the training procedure as a minimax game between the generative and discriminative models.

However, in discrete settings a number of important limitations have prevented major breakthroughs. A major issue involves the complexity of backpropagating the gradients for the generative model. To bypass this differentiation problem, Yu et al. (2016); Che et al. (2017) proposed a cost function inspired by reinforcement learning. The discriminator is used as a reward function, and the generator is trained via policy gradient (Sutton et al. (2000)).

Our approach leverages recent work on discrete GANs. Bose et al. (2018) describes an approach for adversarial contrastive estimation, where discrete GANs are leveraged to implement an adversarial negative sampler. However, our approach differs as in our method, both the generator and the discriminator are trained to work cooperatively. Also, we use a different approach for training the generator, which we call *cooperative importance sampling*, which is based on a multivariate Bernouilli distribution.

3 Our approach

3.1 Concepts and notation

We begin this section by formally defining the task. We denote by $\mathbb{I}$ and $\mathbb{J}$ two sets of objects. These objects can potentially represent sets of users, items or words. Given an $i \in \mathbb{I}$ sampled uniformly from $\mathbb{I}$, we denote by P_i the unknown conditional generative process that generates the set of $j \in \mathbb{J}$ and by X the resulting joint distribution over $\mathbb{I} \times \mathbb{J}$. In our data, we assume that we only observe a finite sample of pairs $(i, j) \sim X$.

We denote $\mathscr{G} = \{G_\theta\}_{\theta \in \Theta}$ as the family of functions, with parameters $\Theta \subset \mathbb{R}^p$. The model G_θ takes an input in $\mathbb{I} \times \mathbb{J}$ and outputs a score $G_\theta(i, j) \in \mathbb{R}$. The objective is to learn a generative model G_θ that fits the conditional process observed in the data.

3.2 P_i modeled as a categorical distribution

Usually, one considers that the items $j \in \mathbb{J}$ are mutually exclusive. Doing so one assumes that the true density P_i of the process X is a categorical distribution defined on $\mathbb{J}$ conditioned on i. This is the implicit model defined when one tries to fit P_i with a softmax distribution g_θ as follows:

$$\forall \mathbf{i} \in I,\ g_\theta(j|\mathbf{i}) = \frac{\exp(G_\theta(i,j))}{\exp(G_\theta(i,j)) + \sum_{j' \in \mathbb{J} \setminus j} \exp(G_\theta(i,j'))} \tag{1}$$

With this model, for a given $i \in I$, one tries to perform Kullback-Leibler divergence minimization between the true distribution P_i and the model of the distribution g_θ:

$$\underset{\theta \in \Theta}{\operatorname{argmin}}\ KL(P_i || g_\theta(.|i)) = \underset{\theta \in \Theta}{\operatorname{argmax}}\ \mathbb{E}_{j \sim P_i}\ [\ln\,(g_\theta(j|i))] \tag{2}$$

$$\underset{\theta \in \Theta}{\operatorname{argmax}}\ \mathbb{E}_{j \sim P_i}\ G_\theta(i,j) - \ln \sum_{j' \in \mathbb{J}} \exp(G_\theta(i,j')) \tag{3}$$

where σ is the sigmoid function.

3.3 P_i MODELED AS MULTIVARIATE BERNOULLI DISTRIBUTION

Currently softmax is used with good results in many problems that fall outside of the classical area of multiclass prediction, such as language modeling, sequence prediction and matrix factorization. We argue that in many of these domains, the *mutual exclusivity* implied in the softmax formulation is overly restrictive and that it could be replaced with a softer version.

Intuition For example, taking a language model task, the probability of seeing in a sentence the word *garden* given the existence of the word *house* does not impact the probability of seeing the word *brick* in the same sentence. We argue that a far more reasonable decomposition would be one that assumes their conditional independence: $P(\textit{garden}, \textit{brick}|\textit{house}) = P(\textit{garden}|\textit{house}) \times P(\textit{brick}|\textit{house})$.

More formally, we have that the joint conditional distribution P_i over $\mathbb{J}$ can be decomposed as a product of a set of Bernoulli distributed random variables over the set $\mathbb{J}$:

$$P_i = P(j_1..j_n|i) = \prod_{x \in \mathbb{J}} P(j_x|i) \tag{4}$$

The problem of mutual exclusivity appears when trying to compute the normalization constant for the conditional probability $P(j|i)$:

$$P(j|i) = \frac{P(i,j)}{P(i,j) + P(i,\neg j)} \tag{5}$$

It is not obvious how to compute $P(i, \neg j)$, e.g. what is the set of events that are mutually exclusive with the pair (i, j). Softmax assumes that all other events are mutual exclusive and we would like to relax this assumption by restricting the set using the input data. To come back to our example, for a given context word i and its target word j, some words are very unlikely given the co-occurence of i and j. For example, one will not see two verbs applied to the same noun or two adverbs describing the same adjective. Therefore, we assume that only words of the same *grammatical category* and *semantic context* will impact their co-occurences with i. However, defining these categories and contexts may be difficult and, as a first version, we simplify this model by assuming that for every context $i \in \mathbb{I}$, there is a set of negatives that will not co-occur with i independently of the observed target word j, meaning that:

$$P(j|i) = \frac{P(i,j)}{P(i,j) + P(i,\neg j)} \approx \frac{P(i,j)}{P(i,j) + P(i,\neg S_i)} \tag{6}$$

where S_i is defined as the true but unknown support set of valid $j's$ for i.

Partially Mutual Exclusive Softmax In order to model each one of the components of P_i we introduce *PME-Softmax*, defined as:

$$\forall (i,j) \in \mathbb{I} \times \mathbb{J},\ g_\theta(j|i) = \frac{\exp(G_\theta(i,j))}{\exp(G_\theta(i,j)) + \sum_{j' \in \neg S_i} \exp(G_\theta(i,j'))} \tag{7}$$

where S_i is the support of P_i. In this formulation, the normalization factor is computed over the positive evidence of the pair (i, j) and the sum of the probabilities over the negative pairs which in theory should sum to zero and it disregards all of the probability mass of i with other $j's$. However, since in reality we do not have access to the support S_i, we replace it with a probabilistic model D_i of $\neg S_i$ and use it to weight all of the sampled pairs (i, j') that are not the current positive pair (i, j). In this way we replace the exact formulation of the *PME-Softmax* with a probabilistic one, where all of the j items are weighted by the probability of being in the negative set:

Probabillistic Partially Mutual Exclusive Softmax

$$\forall (i,j) \in \mathbb{I} \times \mathbb{J},\ g_\theta(j|i) = \frac{\exp(G_\theta(i,j))}{\exp(G_\theta(i,j)) + \sum_{j' \in J \setminus j} \exp(G_\theta(i,j')) \times D_i(j')} \tag{8}$$

where D_i is the probabilistic model of $\neg S_i$.

3.4 THE COOPERATIVE GAME: MODELING D_i

In order to have a model of these negatives, we leverage recent work on GANs by training simultaneously a generator and a discriminator.

We denote $\mathscr{D} = \{D_\alpha\}_{\alpha \in A}$ as the family of functions, with parameters $A \subset \mathbb{R}^p$. The model D_α takes an input $(i, j) \in (\mathbb{I}, \mathbb{J})$ and outputs a score $D_\alpha(i, j) \in \mathbb{R}$. Using the usual GAN setting, D can be trained with a binary cross-entropy loss function (BCE), to discriminate between positive data sampled from X and negatives sampled by our generator.

$$\underset{\alpha \in A}{\text{argmax}} \ \mathbb{E}_{(i,j)\sim X} \ \ln \ \sigma(D_\alpha(i, j)) + \sum_{j=1}^{k} \mathbb{E}_{j \sim g_\theta(.|i)} \ [\ln \ \sigma(-D_\alpha(i, j))] \tag{9}$$

where σ is the sigmoid function.

In our sampling model, for a given $i \in \mathbb{I}$, we use the modeled softmax conditional distribution $g_\theta(.|i)$ to sample negatives. It is true that sampling with g_θ requires computing the score for every possible continuation, but it is an effective way to sample negatives that are close the decision boundary. We define the set of negative items as $\{j_1, ..., j_n \sim g_\theta(.|i)\}$. However, as the model improves during training, the likelihood of sampling positives gets higher. To avoid this possibility, we use D to reweight the negative samples that are false negatives. Intuitively, by doing so, we sample true negatives close to the decision boundary.

This model called *cooperative importance sampling*, has the following training loss:

$$\underset{\theta \in \Theta}{\text{argmax}} \ \mathbb{E}_{(i,j)\sim X} \ G_\theta(i, j) - \ln \left(\sum_{j \in V} \exp(G_\theta(i, j)) \times \sigma(-D(i, j_k)) \right) \tag{10}$$

$$\equiv \underset{\theta \in \Theta}{\text{argmax}} \ \mathbb{E}_{(i,j)\sim X} \ \ln(p_{\theta,\alpha}\big|_V(j|i)) \tag{11}$$

where $V := \{j\} \bigcup \{j_1, ..., j_n\}$, $j_1, ..., j_n \sim g_\theta(.|i)$ and $p_{\theta,\alpha}\big|_V(j|i) = \frac{\exp(G_\theta(i,j)) \times D_\alpha(i,j)}{\sum_{j \in V} \exp(G_\theta(i,j)) \times D_\alpha(i,j)}$, a distribution whose support lies in V.

Improved gradient via g and D Intuitively, we do not use G and D to perform a biased importance sampling approximating softmax. On the contrary, by reweighting our samples with D, we are trying to mimic the true distribution negatives with $p_{\theta,\alpha}\big|_V$. When deriving the sampled softmax loss, we can notice that for a given $(i, j) \in \mathbb{I} \times \mathbb{J}$ and V, the union set of our negatives and the target, the gradient is:

$$\nabla G_\theta(i, j) - \sum_{(i,j) \in V} p_{\theta,\alpha}\big|_V(j|i) \ \nabla G_\theta(i, j) \tag{12}$$

We see that is really important to sample *hard negatives* where the gradient of G is still meaningful. If, for example, the negatives sampled are too easily distinguishable from real data, then they will be far from the decision boundary, have low gradients and therefore will not enable G_θ to improve itself. Also, D guarantees us to oversample true negatives in V.

3.5 ALGORITHM

A description of our algorithm can be found in Algorithm 1. We empirically show improvements with this procedure in the following section.

4 EXPERIMENTS

We compare our *CIS* negative sampling scheme against full softmax and the negative schemes listed below on three types of datasets, namely a 2D artifical dataset where we know the true support, a text datasets on which we evaluate word ranking, word similarity and word analogy and five matrix factorization datasets, on which we evaluate precision@k. Negative sampling baselines: Full Softmax (FS), Uniform sampling (UniS), Popularity based sampling (PopS) and Selfplay (SP).

Algorithm 1 AIS

Require: generator G_θ; discriminator D_α; a generative process X.
Initialize G_θ, D_α with random weights θ, α.

repeat
  for g-steps **do**
    Sample pairs $(i, j) \sim X$
    Generate samples from the softmax distribution $g_\theta(.|i)$
    Train the generator G_θ with Eq. (10)
  end for
  for d-steps **do**
    Sample pairs $(i, j) \sim X$
    Generate samples $(j_1, ..., j_n)$ from $g_\theta(.|i)$
    Train the discriminator D_α with Eq. (9)
  end for
until AIS converges

Method	Blobs ($r_1 = 0.25$)	Blobs ($r_2 = 0.60$)	Blobs ($r_3 = 0.75$)	Swiss Roll
FS	69.0 ± 0.2	77.0 ± 0.2	84.5 ± 0.3	89.1 ± 0.1
UniS	72.3 ± 0.1	70.1 ± 0.1	75.1 ± 0.2	87 ± 0.2
PopS	75.2 ± 0.2	65.0 ± 0.1	77.2 ± 0.1	88.2 ± 0.1
Selfplay	75.7 ± 0.3	76.5 ± 0.1	84.5 ± 0.1	88.5 ± 0.1
CIS	$\mathbf{78.5 \pm 0.1}$	$\mathbf{86.1 \pm 0.2}$	$\mathbf{89.5 \pm 0.1}$	$\mathbf{90.2 \pm 0.1}$

Table 1: Table showing the Accuracy results for the different training methods on the different synthetic datasets with different ratios of positive data. CIS outperforms other models on every dataset but by a smaller margin when the underlying manifold is easier to learn the Swiss-Roll dataset.

The Selfplay method is defined by sampling the negative samples in the softmax distribution g_θ of our model like in our *cooperative importance sampling scheme* except that we do not correct the weights with the help of the discriminator. Sampling within its own distribution allows us to sample negatives close to the decision boundary. However, without D's reweighting, we expect the Selfplay method to perform worse than the CIS model which has been the case in most empirical results.

4.1 Learning 2D discrete distribution: Synthetic datasets

To verify our intuition around true support modeling, we defined a set of 2D artificial datasets where all positive pairs live in 2D shapes and the task of the Word2Vec is to learn these shapes from samples of positives and unlabeled data. The 2D shapes we experimented with are the *Swiss Roll*, the *S Curve* and overlapping blobs of various size. An advantage of the blobs-based simulations is that they allow us to better control the ratio r of positive/negative in our space $(I \times J)$. One can look in the Appendix, where we present different figures showing how well the different methods learn the underlying distribution.

In terms of quantitative results, we report the accuracy of our predictions, since inherently the task is to classify pairs in positives and negatives and at test time we do have access to all of the negative pairs. The results are reported in the Table 1 below. We observe that our cooperative importance sampling is systematically better than other sampling methods and on par with full softmax. Furthermore, it is interesting to note that when the underlying distribution is quite easy to learn the impact of CIS is smaller, as D cannot fully correct the sampling of positive data.

4.2 Language Modeling Tasks

For the language modeling tasks, we ran experiments on the *text8* dataset, a textual data excerpt from Wikipedia taken from *Matt Mohaney's* page. We ran two experiments where we only kept respectively the 12,000 and 30,000 most-occuring words in the data and ran Skip-Gram Word2Vec with a window size 1 and the versions of negative sampling listed above. We ran experiments on

threee different tasks: next word prediction, similarity task and analogy task. We also compared to the FastText (FT) model implemented by Joulin et al. (2016), which is considered to provide state-of-the-art results. We trained it on the same data and benched on different parameters.

Next word prediction In the next word prediction task, we use the model to predict the target word for a given context. We report both the Mean Percentile Rank (MPR) and Precision@1 metrics. We see that on these two metrics our model beats the different sampling methods, including full softmax.

Method	Text8 (12000)		Text8 (30000)	
FS	85.1 ± 0.08	7.2 ± 0.15	89.4 ± 0.08	7.5 ± 0.15
UniS	87.4 ± 0.07	6.0 ± 0.12	90.6 ± 0.07	5.5 ± 0.17
PopS	87.6 ± 0.06	7.2 ± 0.13	90.7 ± 0.06	7.1 ± 0.18
Selfplay	88.2 ± 0.04	8.9 ± 0.10	88.7 ± 0.05	8.6 ± 0.16
CIS	**88.7 $\pm$ 0.08**	**8.9$\pm$ 0.10**	**91.6 $\pm$ 0.08**	**9.3 $\pm$ 0.09**

Table 2: Table showing the MPR and P@1 for the different models on the two text datasets. We see that even using the selfplay method helps us sample hard negatives.

The words similarity task In terms of qualitative results, we show in Table 3 a set of 6 query words and their 6-nearest neighbours in terms of the cosine distance in the embedding space in the 30k different words dataset. We can see that the full softmax and our cooperative sampling method have some similarities in terms of results, except that softmax sometimes lacks coherence, especially as we increase this number of neighbors. On the other hand, FastText and our importance cooperative sampling tend to have very different results. More similarities between words are shown in Appendix.

Method	CIS	FastText	Full Softmax
assembly	senate, legislature parliament, seats cabinet, democracy	assembl, assemblies assemble, assembling assembler, assembled	government, system union, party council, parliament
author	poet, writer composer, historian philosopher, novelist	authored, writer authors, novelist authorship, poet	writer, actor scholar, poet artist, actress
australia	canada, scotia usa, indonesia bulgaria, argentina	australis, australiasia australians, australasian zealand, queensland	island, canada day, georgia kingdom, party
true	meaning, essence false, truth sense, existence	untrue, truth false, truths theistic, falsehood	meaning, case possible, definition or, understanding
essay	bibliography, thinking essays, prophecy story, poetry	essays, essai essayist essayists critique, treatises	treatise, downloads interviews, mentor essays, pupil
final	previous, latest next, first last, second	finale, finals finalized, penultimate last, eventual	previous, last first, another initial, second

Table 3: In this table, we show qualitative results for the Similarity task results. We can see that FastText focuses more on N-grams similarities and shows less diversity.

The word analogy task The goal of the word analogy task is to compare the different embedding structures. With the pair based method, given word i and its analogy j, one has to find, given a third word i', its analogy j'. Usually, we consider two types of analogy tasks: *semantic* and *syntactic* analogies. As a semantic analogy example, suppose we are given the transformation "man" to "woman" and a third word "king", one has to reconstruct "queen". As a syntactic analogy example, suppose we are given the transformation "run" to "runs" and a third word "swim", one has to reconstruct "swims". In terms of quantitative results we show the Prec@1, Prec@5 and Prec@15 in the Table 6 of *CIS* against the baseline models and see a clear uplift in terms of quality. As FastText relies on the n-gram information, they manage to beat CIS on the syntactic analogy task, especially on the 30k different words dataset.

12,000 most-occuring words

Method	**CIS**	**SP**	FT	FS
Semantic				
P@1	**14.8**	9.3	0.0	1.1
P@5	**19.7**	14.3	4.4	7.1
P@15	**21.4**	18.7	17.0	11.0
Syntactic				
P@1	1.1	**1.7**	0.6	0.1
P@5	4.2	**6.8**	1.7	1.7
P@15	6.8	**10.2**	9.3	3.8

30,000 most-occuring words

Method	**CIS**	**SP**	FT	FS
Semantic				
P@1	**14.75**	10.8	1.9	10.7
P@5	20.6	**22.2**	17.0	16.0
P@15	35.8	35.3	**41.17**	21.9
Syntactic				
P@1	1.9	2.7	**8.9**	1.2
P@5	7.15	8.9	**26.7**	5.1
P@15	11.80	16.1	**46.0**	8.7

Table 6: On this table we present the results for the Analogy task on respectively the 12k and 30k different words datasets. On both datasets, CIS/SP are ahead in the Semantic tasks on the Prec@1/5 metrics, as sampling helps us be really performant for the closest neighbors. However, when increasing the number of Nearest Neighbors, FastText cacthes up which might indicate a better structure in the tail.

4.3 Matrix Factorization Experiments : Shopping Baskets and Movies datasets

For the matrix factorization application we performed our experiments on five different public datasets: two shopping basket focused datasets (the *Belgian Retail* dataset and the *UK retail* dataset with respectively 16,470 and 4,070 different items), two recommandation datasets focused on movies (the *Movielens* and the *Netflix* datasets where we only kept respectively 17,128 and 17770 movies that had been ranked by a user respectively over 4 and 4.5 stars).

On these datasets, we report the Prec@1 on the test data as shown in Table 7. We observe that CIS outperforms all baselines. It seems that the model gets relatively better when the vocabulary size of the dataset increases.

Method	Belgian	UK	Movielens	Netflix
FS	10.8 ± 0.1	2.7 ± 0.1	2.2 ± 0.1	2.2 ± 0.1
UniS	10.2 ± 0.2	2.4 ± 0.1	1.4 ± 0.3	1.4 ± 0.2
PopS	10.6 ± 0.1	2.4 ± 0.2	2.5 ± 0.1	1.4 ± 0.2
Selfplay	10.8 ± 0.1	2.5 ± 0.1	2.5 ± 0.1	2.2 ± 0.1
CIS	$\mathbf{11.2 \pm 0.2}$	$\mathbf{3.1 \pm 0.1}$	$\mathbf{3.7 \pm 0.1}$	$\mathbf{2.6 \pm 0.1}$

Table 7: This table shows the P@1 for the 4 different models (Full Softmax, Uniform Sampling, Popularity Sampling, Selfplay and CIS) on the real datasets on the Item-Items task. On the task of Prec@1, CIS consistently outperfoms the full softmax.

5 Conclusions

In this paper, we have proposed *Partially Mutual Exclusive Softmax*, a relaxed version of the full softmax that is more suited in cases with no explicit negatives, e.g., in cases with positive and unlabeled data. In order to model the new softmax we proposed a cooperative negative sampling algorithm. Based on recent progress made on GANs in discrete data settings, our cooperative training approach can be easily applied to models that use standard sampled softmax training, where the generator and discriminator can be of the same family of models. In future work we will investigate the effectiveness of this training procedure on more complex models, and also try to make our mutually exclusive set model more contextual and dependent on both objects i and j within a pair. For example, for a given pair context/target, one might want to use the closest neighbors of the target in the embedding space as negatives. This could enable us to obtain a negative distribution that fits both the context and the target.

REFERENCES

Yoshua Bengio and Jean-Sébastien Senécal. Adaptive importance sampling to accelerate training of a neural probabilistic language model. *IEEE Transactions on Neural Networks*, 19(4):713–722, 2008.

Yoshua Bengio, Jean-Sébastien Senécal, et al. Quick training of probabilistic neural nets by importance sampling. In *AISTATS*, pp. 1–9, 2003.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *CoRR*, abs/1607.04606, 2016. URL `http://arxiv.org/abs/1607.04606`.

Avishek Joey Bose, Huan Ling, and Yanshuai Cao. Adversarial contrastive estimation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1021–1032. Association for Computational Linguistics, 2018.

Tong Che, Yanran Li, Ruixiang Zhang, R. Devon Hjelm, Wenjie Li, Yangqiu Song, and Yoshua Bengio. Maximum-likelihood augmented discrete generative adversarial networks. *CoRR*, abs/1702.07983, 2017. URL `http://arxiv.org/abs/1702.07983`.

Long Chen, Fajie Yuan, Joemon M Jose, and Weinan Zhang. Improving negative sampling for word representation using self-embedded features. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 99–107. ACM, 2018.

Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 213–220. ACM, 2008.

I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and J. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc., Red Hook, 2014.

M. Gutmann and A. Hyvärinen. *Noise-contrastive estimation: A new estimation principle for unnormalized statistical models*, volume 9 of *JMLR WCP*, pp. 297–304. Journal of Machine Learning Research - Proceedings Track, 2010.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. *CoRR*, abs/1412.2007, 2014. URL `http://arxiv.org/abs/1412.2007`.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759, 2016.

Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802, 2016. URL `http://arxiv.org/abs/1609.04802`.

Wee Sun Lee and Bing Liu. Learning with positive and unlabeled examples using weighted logistic regression. In *ICML*, volume 3, pp. 448–455, 2003.

Xiaoli Li and Bing Liu. Learning to classify texts using positive and unlabeled data. In *IJCAI*, volume 3, pp. 587–592, 2003.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013b.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *CoRR*, abs/1802.05957, 2018. URL http://arxiv.org/abs/1802.05957.

Andriy Mnih and Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models. *arXiv preprint arXiv:1206.6426*, 2012.

Ulrich Paquet and Noam Koenigstein. One-class collaborative filtering with random graphs. In *Proceedings of the 22nd international conference on World Wide Web*, pp. 999–1008. ACM, 2013.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015. URL http://arxiv.org/abs/1511.06434.

Yafeng Ren, Donghong Ji, and Hongbin Zhang. Positive unlabeled learning for deceptive reviews detection. In *EMNLP*, pp. 488–498, 2014.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pp. 1057–1063, 2000.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. *CoRR*, abs/1609.05473, 2016. URL http://arxiv.org/abs/1609.05473.

A EXPERIMENTS

A.1 LEARNING 2D DISCRETE DISTRIBUTION

Here we show the different heatmaps regarding the different conditional models learned by the model on 4 different methods: Popularity sampling, Selfplay sampling, Full Softmax and our Cooperative Importance Sampling. See Image 1 and Image 2.

A.2 THE WORDS SIMILARITY TASK

Here we present some other similarities between words generated from three different methods: Full Softmax, FastText and our CIS model. See Figure 8.

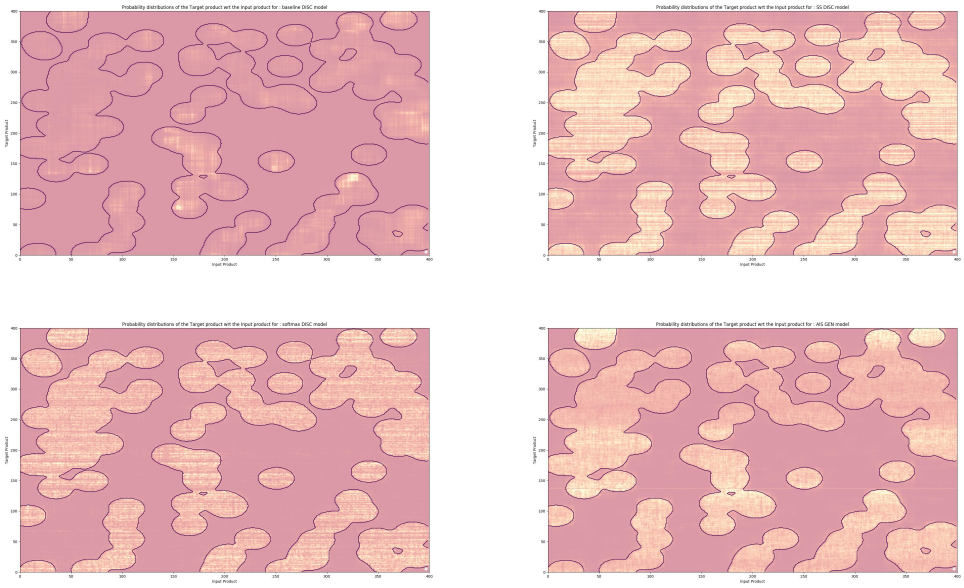

Figure 1: Learning the conditional distribution for one of the blobs dataset $r_2 = 0.6$. From the top left to the bottom right: the Popularity based sampling, the Selfplay model (top left) the Softmax and the CIS model. We see the impact of the CIS model on the learning of the right conditional distribution as the distinction between true data (inside the blobs) and negative data is clearer.

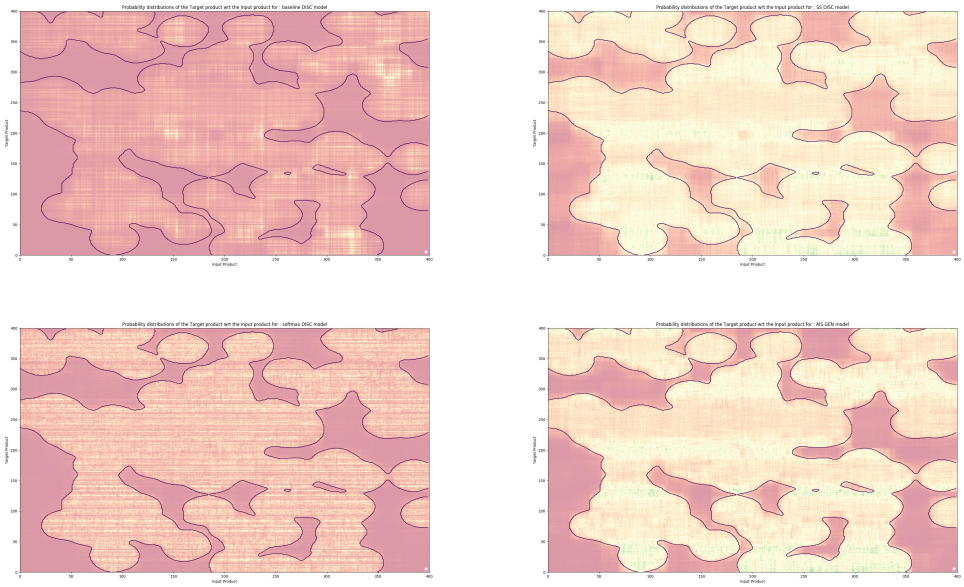

Figure 2: Learning the conditional distribution for one of the blobs dataset $r_3 = 0.75$. From the top left to the bottom right: the Popularity based sampling, the Selfplay model (top left) the Softmax and the CIS model. Again, on this image, we see the impact of the CIS model on the learning of the right conditional distribution as the distinction between true data (inside the blobs) and negative data is clearer.

Method	AIS	FastText	Full Softmax
animals	bacteria, humans beings, structures organisms, plants	animal, animalia mammals, humans vertebrates, carnivores	methods, species techniques, practices things, plants
brother	son, grandson daughter, father mother, uncle	brothers, broth brotherhood, father father, mother	son, b countess, maria prince, nephew
charles	robert, henry edward, elizabeth peter, james	charley, charleston charlie, charlotte frederick, louise	henry, erasmus peter, thomas james, john
constantinople	monarch, prussia ruins, athens prague, antioch	constantine, constantinus constantin, adrianople nicaea, chalcedon	carthage, emperors reformation, dynasty ks, bohemia
freedom	hate, sociology acceptance, resistance conscious, subject	freedoms, freed freedmen, edom liberty, free	free, learning fairness, anymore huygens, technology

Table 8: Table showing results for the Similarity task results : in some cases we can see that the CIS model does reflect more the meaning of the word.