Notes on Noise Contrastive Estimation and Negative Sampling

Chris Dyer
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA, 15213
cdyer@cs.cmu.edu

Abstract

Estimating the parameters of probabilistic models of language such as maxent models and probabilistic neural models is computationally difficult since it involves evaluating partition functions by summing over an entire vocabulary, which may be millions of word types in size. Two closely related strategies—**noise contrastive estimation** (Mnih and Teh, 2012; Mnih and Kavukcuoglu, 2013; Vaswani et al., 2013) and **negative sampling** (Mikolov et al., 2012; Goldberg and Levy, 2014)—have emerged as popular solutions to this computational problem, but some confusion remains as to which is more appropriate and when. This document explicates their relationships to each other and to other estimation techniques. The analysis shows that, although they are superficially similar, NCE is a general parameter estimation technique that is asymptotically unbiased, while negative sampling is best understood as a family of binary classification models that are useful for learning word representations but not as a general-purpose estimator.

1 Introduction

Let us assume the following model of language which predicts a word w in a vocabulary V based on some given context c:[1]

$$p_\theta(w \mid c) = \frac{u_\theta(w, c)}{\sum_{w' \in V} u_\theta(w', c)} = \frac{u_\theta(w, c)}{Z_\theta(c)}, \tag{1}$$

where $u_\theta(w, c) = \exp s_\theta(w, c)$ assigns a score to a word in context, $Z(c)$ is the partition function that normalizes this into a probability distribution, and $s_\theta(w, c)$ is differentiable with respect to θ. The standard learning procedure is to maximize the likelihood of a sample of training data. Unfortunately, computing this probability (and its derivatives) is expensive since this requires summing over all words in V, which is generally very large.

What can be done? Since the derivatives of the log likelihood include terms that are the expected values of the parameters under the model distribution, the classic strategy has been to use **importance sampling** and related Monte Carlo techniques to approximate these expectations (Bengio and Senécal, 2003). Noise contrastive estimation and negative sampling represent an evolution of these techniques. These work by transforming the computationally expensive learning problem into a binary classification *proxy problem* that uses the same parameters but requires statistics that are easier to compute.

[1] By *language model* I mean a model that generates one word at a time, conditional on any other ambient context such as previously generated or surrounding words, a topic label, text in another language, etc. Excluded are so-called "whole-sentence" or "globally normalized" language models. While these can also, in principle, be learned using the techniques described in these notes, this exposition focuses on models that predict a single word at a time.

1.1 Empirical distributions, Noise distributions, and Model distributions

I will refer to $\tilde{p}(w \mid c)$ and $\tilde{p}(c)$ as empirical distributions. Our task is to find the parameters θ of a model $p_\theta(w \mid c)$ that approximates the empirical distribution as closely as possible, in terms of minimal cross-entropy. To avoid costly summations, a "noise" distribution, $q(w)$, is used. In practice q is a uniform, empirical unigram, or "flattened" empirical unigram distribution (by exponentiating each probability by $0 < \alpha < 1$ and renormalizing).

2 Noise contrastive estimation (NCE)

NCE reduces the language model estimation problem to the problem of estimating the parameters of a probabilistic binary classifier that uses the same parameters to distinguish samples from the empirical distribution from samples generated by the noise distribution (Gutmann and Hyvärinen, 2010). The two-class training data is generated as follows: sample a c from $\tilde{p}(c)$, then sample one "true" sample from $\tilde{p}(w \mid c)$, with auxiliary label $D = 1$ indicating the datapoint is drawn from the true distribution, and k "noise" samples from q, with auxiliary label $D = 0$ indicating these data points are noise. Thus, given c, the joint probability of (d, w) in the two-class data has the form of the mixture of two distributions:

$$p(d, w \mid c) = \begin{cases} \frac{k}{1+k} \times q(w) & \text{if } d = 0 \\ \frac{1}{1+k} \times \tilde{p}(w \mid c) & \text{if } d = 1 \end{cases}.$$

Using the definition of conditional probability, this can be turned into a conditional probability of d having observed w and c:

$$\begin{aligned} p(D = 0 \mid c, w) &= \frac{\frac{k}{1+k} \times q(w)}{\frac{1}{1+k} \times \tilde{p}(w \mid c) + \frac{k}{1+k} \times q(w)} \\ &= \frac{k \times q(w)}{\tilde{p}(w \mid c) + k \times q(w)} \\ p(D = 1 \mid c, w) &= \frac{\tilde{p}(w \mid c)}{\tilde{p}(w \mid c) + k \times q(w)}. \end{aligned}$$

Note that these probabilities are written in terms of the empirical distribution.

NCE replaces the empirical distribution $\tilde{p}(w \mid c)$ with the model distribution $p_\theta(w \mid c)$, and θ is chosen to maximize the conditional likelihood of the "proxy corpus" created as described above. But, thus far, we have not solved any computational problem yet: $p_\theta(w \mid c)$ still requires evaluating the partition function—all we have done is transform the objective by adding some noise. To avoid the expense of evaluating the partition function, NCE makes two further assumptions. First, it proposes partition function value $Z(c)$ be estimated as parameter z_c (thus, for every empirical c, classic NCE introduces one parameter). Second, for neural networks with lots of parameters, it turns out that fixing $z_c = 1$ for all c is effective (Mnih and Teh, 2012). This latter assumption both reduces the number of parameters and encourages the model to have "self-normalized" outputs (i.e., $Z(c) \approx 1$). Making these assumptions, we can now write the conditional likelihood of being a noise sample or true distribution sample in terms of θ as

$$\begin{aligned} p(D = 0 \mid c, w) &= \frac{k \times q(w)}{u_\theta(w, c) + k \times q(w)} \\ p(D = 1 \mid c, w) &= \frac{u_\theta(w, c)}{u_\theta(w, c) + k \times q(w)}. \end{aligned}$$

We now have a binary classification problem with parameters θ that can be trained to maximize conditional

log-likelihood of $\mathcal{D}$, with k negative samples chosen:

$$\mathcal{L}_{\text{NCE}_k} = \sum_{(w,c)\in\mathcal{D}} \left(\log p(D = 1 \mid c, w) + k\mathbb{E}_{\overline{w}\sim q} \log p(D = 0 \mid c, \overline{w})\right).$$

Unfortunately, the expectation of the second term in this summation is still a difficult summation—it is k times the expected log probability (according to the current model) of producing a negative label under the noise distribution over all words in V in a context c. We still have a loop over the entire vocabulary. The final step is therefore to replace this expectation with its Monte Carlo approximation:

$$\begin{aligned}\mathcal{L}^{\text{MC}}_{\text{NCE}_k} &= \sum_{(w,c)\in\mathcal{D}} \left(\log p(D = 1 \mid c, w) + k \times \sum_{i=1,\overline{w}\sim q}^{k} \frac{1}{k} \times \log p(D = 0 \mid c, \overline{w})\right) \\ &= \sum_{(w,c)\in\mathcal{D}} \left(\log p(D = 1 \mid c, w) + \sum_{i=1,\overline{w}\sim q}^{k} \log p(D = 0 \mid c, \overline{w})\right).\end{aligned}$$

2.1 Asymptotic analysis

Although the objective $\mathcal{L}_{\text{NCE}_k}$ is intractable, its derivative sheds light on why NCE works. This quantity may be written as

$$\frac{\partial}{\partial\theta}\mathcal{L}_{\text{NCE}_k} = \sum_{(w',c)\in\mathcal{D}} \left(\sum_{w\in V} \frac{k \times q(w)}{u_\theta(w \mid c) + k \times q(w)} \times (\tilde{p}(w \mid c) - u_\theta(w \mid c)) \frac{\partial}{\partial\theta} \log u_\theta(w \mid c)\right).$$

It is easy to see that in the limiting case as $k \to \infty$, this derivative tends to the gradient of the log likelihood of $\mathcal{D}$ under p_θ (and furthermore, $\mathcal{L}^{\text{MC}}_{\text{NCE}_k} \to \mathcal{L}_{\text{NCE}_k}$). That is, the gradient is 0 when the model distribution matches the empirical distribution.

3 Negative sampling

Negative sampling is a variation of NCE used by the popular `word2vec` tool which generates a proxy corpus and also learns θ as a binary classification problem, but it defines the conditional probabilities given (w, c) differently:

$$\begin{aligned}p(D = 0 \mid c, w) &= \frac{1}{u_\theta(w, c) + 1} \\ p(D = 1 \mid c, w) &= \frac{u_\theta(w, c)}{u_\theta(w, c) + 1}.\end{aligned}$$

This objective can be understood in several ways. First, it is equivalent to NCE when $k = |V|$ and q is uniform. Second, it can be understood as the hinge objective of Collobert et al. (2011) where the $\max$ function has been replaced with a softmax. As a result, aside from the $k = |V|$ and uniform q case, the conditional probabilities of D given (w, c) are not consistent with the language model probabilities of (w, c) and therefore the θ estimated using this as an objective will not optimize the likelihood of the language model in Eq. 1. Thus, while negative sampling may be appropriate for word representation learning, it does not have the same asymptotic consistency guarantees that NCE has.

4 Conclusion

NCE is an effective way of learning parameters for an arbitrary locally normalized language model. However, negative sampling should be thought of as an alternative task for generating representations of words

for use in other tasks, but not, itself, as a method for learning parameters in a generative model of language. Thus, if your goal is language modeling, you should use NCE; if your goal is word representation learning, you should consider both NCE and negative sampling.

References

Yoshua Bengio and Jean-Sébastien Senécal. 2003. Quick training of probabilistic neural nets by importance sampling. In *Proc. AISTATS*.

Ronan Collobert, Jason Weston, León Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *JMLR*.

Yoav Goldberg and Omer Levy. 2014. `word2vec` explained: Deriving Mikolov et al.'s negative-sampling word-embedding method.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proc. AISTATS*.

Tomas Mikolov, Ilya Sutskeve, Kai Chen, Greg Corrado, and Jeffrey Dean. 2012. Distributed representations of words and phrases and their compositionality. In *Proc. NIPS*.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Proc. NIPS*.

Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. In *Proc. ICML*.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation. In *Proc. EMNLP*.