Notes on Noise Contrastive Estimation (NCE)

David Meyer

dmm@{1-4-5.net,uoregon.edu,...}

March 10, 2017

1 Introduction

In this note we follow the notation used in [2]. Suppose $\mathcal{X} = (x_1, x_2, \cdots, x_{T_d})$ is a sample of a random vector $x \in \mathbb{R}^n$, where each x_i is drawn *unknown* probability density function p_d. Now, one way to describe the properties of the observed data $\mathcal{X}$ is to describe its properties relative to the properties of some reference data $\mathcal{Y}$. The basic idea behind Noise Contrastive Estimation (NCE) is to draw the reference (noise) sample $\mathcal{Y} = (y_1, y_2, \cdots, y_{T_n})$ from a known pdf p_n and then estimate the ratio p_d/p_n, which will in turn give us an estimate of p_d.

2 Definitions

Let $\mathcal{U} = \mathcal{X} \cup \mathcal{Y} = \{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_{T_d+T_n}\}$, where T_d is the number of data samples and T_n is the number of samples from a noise distribution.[1] We associate with each datapoint $\mathbf{u}_t$ a class label C_t such that

$$C_t = \begin{cases} 1 & \text{if } \mathbf{u}_t \in \mathcal{X} \\ 0 & \text{if } \mathbf{u}_t \in \mathcal{Y} \end{cases}$$

Since p_d is unknown, we model $p(.|C = 1) = p_m(.; \theta)$. Note that we're making an assumption here that there exists a $\boldsymbol{\theta}^*$ such that $p_d(.) = p_m(.; \boldsymbol{\theta}^*)$. That is, we assume that the empirical distribution $p_d(.)$ is a member of the parameterized family $p_m(.; \boldsymbol{\theta})$.

Given these definitions, the likelihoods are

[1] Some authors characterize the problem as drawing 1 sample from the empirical distribution ($Td = 1$) and k samples from the reference (noise) distribution ($T_n = k$), where the total number of samples per "turn" is $k + 1$. See e.g., https://arxiv.org/pdf/1410.8251.pdf

$$p(\mathbf{u}|C=1) = p_m(\mathbf{u};\theta) \qquad \#\text{ data} \tag{1}$$
$$p(\mathbf{u}|C=0) = p_n(\mathbf{u}) \qquad \#\text{ noise} \tag{2}$$

The priors are

$$P(C=1) = \frac{T_d}{T_d+T_n} \tag{3}$$
$$P(C=0) = \frac{T_n}{T_d+T_n} \tag{4}$$

The probability of any given $\mathbf{u}$, $P(\mathbf{u})$, is thus

$$P(\mathbf{u}) = (P(C=1) * p(\mathbf{u}|C=1)) + (P(C=0) * p(\mathbf{u}|C=0)) \tag{5}$$
$$= \left(\frac{T_d}{T_d+T_n} * p_m(\mathbf{u};\theta)\right) + \left(\frac{T_n}{T_d+T_n} * p_n(\mathbf{u})\right) \tag{6}$$

Remembering Bayes' Rule:

$$P(\Theta|\mathcal{X}) = \frac{P(\mathcal{X}|\Theta)P(\Theta)}{P(\mathcal{X})} \qquad \#\text{ posterior} = \frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}}$$

we get the posterior probabilities

$$P(C=1|\mathbf{u};\theta) = \frac{P(\mathbf{u}|C=1;\theta) * P(C=1)}{P(\mathbf{u})} \tag{7}$$
$$= \frac{p_m(\mathbf{u};\theta) * \frac{T_d}{T_d+T_n}}{\left(\frac{T_d}{T_d+T_n} * p_m(\mathbf{u};\theta)\right) + \left(\frac{T_n}{T_d+T_n} * p_n(\mathbf{u})\right)} \tag{8}$$
$$= \frac{p_m(\mathbf{u};\theta)}{\left(\left(\frac{T_d}{T_d+T_n} * p_m(\mathbf{u};\theta)\right) + \left(\frac{T_n}{T_d+T_n} * p_n(\mathbf{u})\right)\right) * \frac{T_d+T_n}{T_d}} \tag{9}$$
$$= \frac{p_m(\mathbf{u};\theta)}{p_m(\mathbf{u};\theta) + (\frac{T_n}{T_d})p_n(\mathbf{u})} \tag{10}$$
$$P(C=1|\mathbf{u};\theta) = \frac{p_m(\mathbf{u};\theta)}{p_m(\mathbf{u};\theta) + vp_n(\mathbf{u})} \qquad \#\ v = \frac{T_n}{T_d} = \frac{P(C=0)}{P(C=1)} \tag{11}$$
$$P(C=0|\mathbf{u};\theta) = \frac{vp_n(\mathbf{u})}{p_m(\mathbf{u};\theta) + vp_n(\mathbf{u})} \qquad \#\text{ same analysis} \tag{12}$$

Now we have

$$P(C=1|\mathbf{u};\theta) = \frac{p_m(\mathbf{u};\theta)}{p_m(\mathbf{u};\theta) + vp_n(\mathbf{u})} \tag{13}$$

$$= \frac{p_m(\mathbf{u};\theta) * \frac{1}{p_m(\mathbf{u};\theta)}}{(p_m(\mathbf{u};\theta) + vp_n(\mathbf{u})) * \frac{1}{p_m(\mathbf{u};\theta)}} \tag{14}$$

$$= \frac{1}{1 + v\frac{p_n(\mathbf{u})}{p_m(\mathbf{u};\theta)}} \tag{15}$$

$$= \Big(1 + v\frac{p_n(\mathbf{u})}{p_m(\mathbf{u};\theta)}\Big)^{-1} \tag{16}$$

This is the first time we see an estimate of ratio we are looking for, namely $\frac{p_n(\mathbf{u})}{p_m(\mathbf{u};\theta)}$. Now, define $G(\mathbf{u};\theta)$

$$G(\mathbf{u};\theta) = \ln \frac{p_m(\mathbf{u};\theta)}{p_n(\mathbf{u})} \tag{17}$$

$$= \ln p_m(\mathbf{u};\theta) - \ln p_n(\mathbf{u}) \tag{18}$$

$G(\mathbf{u};\theta)$ is called the *log-ratio* between $p_m(\mathbf{u};\theta)$ and $p_n(\mathbf{u})$. If we let $P(C=1|\mathbf{u};\theta) = h(\mathbf{u};\theta)$ then $h(\mathbf{u};\theta)$ can be written as

$$h(\mathbf{u};\theta) = r_v(G(\mathbf{u};\theta)) \tag{19}$$

where

$$r_v(u) = \frac{1}{1 + v\exp(-u)} \tag{20}$$

that is, $r_v(.)$ is the sigmoid/logistic function parameterized by v.

Note that the class labels C_t are assumed to be Bernoulli and iid. The *conditional log-likelihood* is given by

$$\ell(\boldsymbol{\theta}) = \sum_{t=1}^{T_d+T_n} C_t \ln P(C_t=1|\mathbf{u}_t;\boldsymbol{\theta}) + (1-C_t)\ln P(C=0|\mathbf{u}_t;\boldsymbol{\theta}) \tag{21}$$

$$= \sum_{t=1}^{T_d} \ln\big[h(\mathbf{x}_t;\boldsymbol{\theta})\big] + \sum_{t=1}^{T_n} \ln\big[1-h(\mathbf{y}_t;\boldsymbol{\theta})\big] \tag{22}$$

Of the many interesting things here: optimizing $\ell(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ leads to an estimate $G(.;\hat{\boldsymbol{\theta}})$ of the log-ratio $\ln(p_d/p_n)$. Said another way, an approximate description of X

relative to Y can be obtained by optimizing Equation 22. Note also that $-\ell(\boldsymbol{\theta})$ is the cross-entropy function.

This result shows that *density estimation*, an unsupervised learning task, can be performed by logistic regression (which is supervised learning).

3 The NCE Estimator

Recall that the normalized pdf $p_m(.;\hat{\theta})$ satisfies the following constraints:

$$\int p_m(\mathbf{u};\boldsymbol{\theta})d\mathbf{u} = 1 \qquad \#\text{ normalization constraint} \tag{23}$$

$$p_m(.;\boldsymbol{\theta}) \geq 0 \qquad \#\text{ positivity constraint} \tag{24}$$

Note that if the constraints for the model $p_m(.;\boldsymbol{\theta})$ hold for all $\boldsymbol{\theta}$ (and not just $\hat{\boldsymbol{\theta}}$) then the model is said to be *normalized* and Maximum Likelihood can be used to estimate $\boldsymbol{\theta}$. If only the positivity constraint (and not the normalization constraint) is satisfied then we say the model is *unnormalized*. The main assumption here is that there exists at least one set of parameters, $\boldsymbol{\theta}^*$, for which the unnormalized model integrations to one[2]. That is, $p_d(.) = p_m(.;\boldsymbol{\theta}^*)$. The unnormalized model is usually denoted by $p_m^0(.;\boldsymbol{\alpha})$.[3]

Now, define the partition function $Z(\alpha)$ as

$$Z(\alpha) = \int p_m^0(\boldsymbol{u};\alpha)d\boldsymbol{u} \tag{25}$$

$Z(\alpha)$ can be used to convert an unnormalized model $p_m^0(.;\alpha)$ into a normalized one: $p_m^0(.;\alpha)/Z(\alpha)$ which integrates to one for every value of α. Examples of distributions that are typically specified by unnormalized models include Gibbs distributions and Markov networks. The problem, however, is that the mapping $\alpha \to Z(\alpha)$ is defined by an integral (Equation 25), so unless $p_m^0(.;\alpha)$ has some convenient form, the integral usually cannot be computed analytically (and thus $Z(\alpha)$ is not available in closed form). For low-dimensional problems numerical integration can provide good accuracy, but for high-dimensional problems computation of $Z(\alpha)$ becomes intractable (cf the curse of dimensionality).

[2]Another way to think about this: as mentioned above, $p_d(.)$ comes from the family of parameterized distributions, $p_m(\mathbf{u};\boldsymbol{\theta})$.

[3]$p_m^0(.;\alpha)$, the unnormalized model, is sometimes written $\phi(\xi;\theta)$ [1]; the usual lack of standardized notation in machine learning....

It is worth noting that unnormalized models are widely used, with examples including models of images (Markov Random Fields), models of text (neural probabilistic language models), various models in physics (Ising models), and more. The advantages of using unnormalized models in that they are often easier to specify than normalized models. The disadvantage is that the likelihood function is generally intractable.

Noise Contrastive Estimation (NCE) is an estimation method for unnormalized models. The basic idea here is to consider Z, or equivalently $c = \ln 1/Z$ as a parameter to the model (rather than a partition function). In particular, NCE considers $p_m^0(.;\alpha)$ to include a new normalizing parameter c and estimates

$$\ln p_m(;.;\boldsymbol{\theta}) = \ln p_m^0(.;\alpha) + c \tag{26}$$

where the parameter $\boldsymbol{\theta} = (\alpha, c)$. In particular, the estimate $\hat{\boldsymbol{\theta}} = (\hat{\alpha}, \hat{c})$ then causes the unnormalized model $(p_m^0(.;\alpha))$ to match the shape of p_d and $\hat{c}$ provides the appropriate scaling so that the normalization constraint (Equation 24) holds.

Observe that after training, $\hat{c}$ provides an estimate for $\ln 1/Z(\alpha)$. Of course, if the model is normalized in the first place c is unnecessary. Given these definitions, we can define the estimator $\hat{\boldsymbol{\theta}}_T$ as the value of $\boldsymbol{\theta}$ which maximizes

$$J_T(\boldsymbol{\theta}) = \frac{1}{T_d}\left\{ \sum_{t=1}^{T_d} \ln\left[h(\mathbf{x}_t;\boldsymbol{\theta})\right] + \sum_{t=1}^{T_n} \ln\left[1 - h(\mathbf{y}_t;\boldsymbol{\theta})\right] \right\} \tag{27}$$

where the nonlinearity $h(.;\boldsymbol{\theta})$ is defined in Equation 19. Note that the objection function J_T is the log-likelihood (Equation 22) divided by T_d. J_T can be rewritten as

$$J_T(\boldsymbol{\theta}) = \frac{1}{T_d} \sum_{t=1}^{T_d} \ln\left[h(\mathbf{x}_t;\boldsymbol{\theta})\right] + v\frac{1}{T_n} \sum_{t=1}^{T_n} \ln\left[1 - h(\mathbf{y}_t;\boldsymbol{\theta})\right] \tag{28}$$

Interestingly, note that $h(.;\boldsymbol{\theta}) \in (0,1)$ and

$$h(.;\boldsymbol{\theta}) = \left\{ \begin{array}{ll} 0 & \lim G(.;\boldsymbol{\theta}) \to -\infty \\ 1 & \lim G(.;\boldsymbol{\theta}) \to \infty \end{array} \right.$$

As a result, the optimal paratement $\hat{\boldsymbol{\theta}}_T$ makes $G(\mathbf{u}_T; \hat{\boldsymbol{\theta}}_T)$ as large as possible for $\mathbf{u}_T \in X$ and as small as possible for $\mathbf{u}_T \in Y$. Said another way, we determine the parameters $\boldsymbol{\theta}$ such that $P(C = 1; \mathbf{u}; \boldsymbol{\theta})$ is large for most $\mathbf{x}_i$ and small for most $\mathbf{y}_i$.

Amazingly, this means that in this case logistic regression has learned to discriminate (classify) between the data and noise sets. That is, what we end up with is unsupervised learning by supervised learning, where successful classification is equivalent to learning the differences between the data and the noise. Here we used nonlinear logistic regression for classification, but other classifiers are possible.

References

[1] Noise-Contrastive Estimation and its Generalizations.

[2] Aapo Hyvarinen Michael U. Gutmann. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 12:307–361, 2012.