Score Matching, adapted from Köster 2009 PhD Thesis

1 Score Matching

Score matching [1, 3, 2] (SM) is a statistical method that allows the estimation of statistical models which can only be determined up to a multiplicative normalization constant. These so-called "energy-based" models come up frequently in machine learning and computational neuroscience. Previously problems of this kind had to be solved with Markov Chain Monte Carlo methods, which are typically very time-consuming.

Consider a random vector $\mathbf{x} \in \mathbb{R}^n$ that follows a pdf $p_{\mathbf{x}}(\boldsymbol{\xi})$. We define a parametrized model density $p(\boldsymbol{\xi}|\boldsymbol{\Theta})$ where $\boldsymbol{\Theta}$ is a parameter vector that we would like to estimate.

For the kind of problem we consider here the normalization constant Z of the pdf cannot be computed, and we use q do denote the unnormalized distribution. In the form of a log-probability we have the model:

$$\log p(\boldsymbol{\xi}|\boldsymbol{\Theta}) = \log q(\boldsymbol{\xi}|\boldsymbol{\Theta}) - \log Z(\boldsymbol{\Theta}) \tag{1}$$

The model score function, which we define as the gradient of the log-probability with respect to the data, is obviously identical for q and p, and given by:

$$\Psi(\boldsymbol{\xi};\boldsymbol{\Theta}) = \nabla_{\boldsymbol{\xi}} \log p(\boldsymbol{\xi};\boldsymbol{\Theta}) \tag{2}$$

Likewise the score function of the observed data is denoted by

$$\Psi_{\mathbf{x}}(.) = \nabla_{\boldsymbol{\xi}} \log p_{\mathbf{x}}(.) \tag{3}$$

Working with the score function thus has the advantage that it is independent of the normalization constant Z. The model can now be estimated by minimizing the squared distance between the *model score function* $\Psi(\xi;\boldsymbol{\Theta})$ and the *data score function* $\Psi_{\mathbf{x}}(.)$ This objective function is defined by

$$J(\boldsymbol{\Theta}) = \frac{1}{2} \int_{\boldsymbol{\xi} \in \mathbb{R}^n} p_{\mathbf{x}}(\boldsymbol{\xi}) \|\Psi(\boldsymbol{\xi};\boldsymbol{\Theta}) - \Psi_{\mathbf{x}}(\boldsymbol{\xi})\|^2 d\boldsymbol{\xi} \tag{4}$$

This may not appear to be very useful at first sight, because estimating the data score function is a nonparametric problem, and would require as much effort as an estimate of the normalization constant.

We will now sketch a proof how a much simpler form of the objective function can be obtained. The full proof can be found in [1]. We start by expanding the squared term to

$$J(\boldsymbol{\Theta}) = \frac{1}{2} \int_{\boldsymbol{\xi} \in \mathbb{R}^n} p_{\mathbf{x}}(\boldsymbol{\xi}) \|\Psi(\boldsymbol{\xi};\boldsymbol{\Theta})\|^2 d\boldsymbol{\xi} + \frac{1}{2} \int_{\boldsymbol{\xi} \in \mathbb{R}^n} p_{\mathbf{x}}(\boldsymbol{\xi}) \|\Psi_{\mathbf{x}}(\boldsymbol{\xi})\|^2 d\boldsymbol{\xi} - \int_{\boldsymbol{\xi} \in \mathbb{R}^n} p_{\mathbf{x}}(\boldsymbol{\xi}) \Psi(\boldsymbol{\xi};\boldsymbol{\Theta})^T \Psi_{\mathbf{x}}(\boldsymbol{\xi}) d\boldsymbol{\xi} \tag{5}$$

Here we note that the first term does not depend on the data score function, so rewriting it with the inner product expanded as a sum we get

$$\frac{1}{2} \int_{\boldsymbol{\xi} \in \mathbb{R}^n} p_{\mathbf{x}}(\boldsymbol{\xi}) \|\Psi(\boldsymbol{\xi};\boldsymbol{\Theta})\|^2 d\boldsymbol{\xi} = \int_{\boldsymbol{\xi} \in \mathbb{R}^n} p_{\mathbf{x}} \sum_{i=1}^{n} \frac{1}{2} \psi_i^2(\boldsymbol{\xi};\boldsymbol{\Theta}) d\boldsymbol{\xi} \tag{6}$$

The second term is constant wrt. $\mathbf{\Theta}$, so we simply set

$$\frac{1}{2}\int_{\boldsymbol{\xi}\in\mathbb{R}^n} p_{\mathbf{x}}(\boldsymbol{\xi})\|\Psi_{\mathbf{x}}(\boldsymbol{\xi})\|^2 d\boldsymbol{\xi} = C \tag{7}$$

Thus we focus on the third term, where we start by writing out the inner product

$$\sum_i \int_{\boldsymbol{\xi}\in\mathbb{R}^n} p_{\mathbf{x}}(\boldsymbol{\xi})\psi_i(\boldsymbol{\xi};\mathbf{\Theta})\psi_{\mathbf{x},i}(\boldsymbol{\xi})d\boldsymbol{\xi} \tag{8}$$

and consider only a single term. We now rewrite the score function $\psi_{\mathbf{x},i}(\boldsymbol{\xi}) = \frac{\partial \log p_x(\boldsymbol{\xi})}{\partial \xi_i}$, so making use of the chain rule, the term becomes

$$\sum_i \int_{\boldsymbol{\xi}\in\mathbb{R}^n} p_{\mathbf{x}}(\boldsymbol{\xi})\psi_i(\boldsymbol{\xi};\mathbf{\Theta})\left[\frac{\partial \log p_x(\boldsymbol{\xi})}{\partial \xi_i}\right] d\boldsymbol{\xi} = \sum_i \int_{\boldsymbol{\xi}\in\mathbb{R}^n} \frac{p_{\mathbf{x}}(\boldsymbol{\xi})}{p_{\mathbf{x}}(\boldsymbol{\xi})}\frac{\partial p_x(\boldsymbol{\xi})}{\partial \xi_i}\psi_i(\boldsymbol{\xi};\mathbf{\Theta})d\boldsymbol{\xi} \tag{9}$$

We then use multivariate partial integration [1] to obtain the i-th term as

$$-\int_{\boldsymbol{\xi}\in\mathbb{R}^n} \frac{\partial p_x(\boldsymbol{\xi})}{\partial \xi_i}\psi_i(\boldsymbol{\xi};\mathbf{\Theta})d\boldsymbol{\xi} = \int_{\boldsymbol{\xi}\in\mathbb{R}^n} \frac{\partial \psi_i(x;\mathbf{\Theta})}{\partial \xi_i} \tag{10}$$

Working with a sample of data, we replace the integrals w.r.t. p_x with sample expectations. Putting this together we obtain the expression

$$\tilde{J}(\mathbf{\Theta}) = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\left[\frac{\partial \psi_i(x(t);\mathbf{\Theta})}{\partial \xi_i} + \frac{1}{2}\psi_i^2(\mathbf{x}(t);\mathbf{\Theta})\right] + C \tag{11}$$

Score matching has been shown to be consistent in [1], so if the data follows the model, the method is guaranteed to converge.

1.1 A Simple Example

Consider the simple problem of fitting the mean and variance of a univariate gaussian to observed data samples $x(t)$. We have the log-likelihood

$$\log p(x(t)|\mu,\sigma^2) = -\frac{1}{2\sigma^2}(x(t)-\mu)^2 - \log Z \tag{12}$$

where the partition function Z is treated as unknown. The score function of the model is

$$\psi = \frac{\partial}{\partial x}\log p = -\frac{1}{\sigma^2}(x(t)-\mu) \tag{13}$$

and the derivative of the score function

$$\psi' = \frac{\partial^2}{\partial x^2}\log p = -\frac{1}{\sigma^2} \tag{14}$$

so the sample version of the score matching objective for T observations is

$$J = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{2}\left(\sigma^{-2}(x(t)-\mu)\right)^2 - \sigma^{-2} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{2}\sigma^{-4}(x(t)-\mu)^2 - \sigma^{-2}. \tag{15}$$

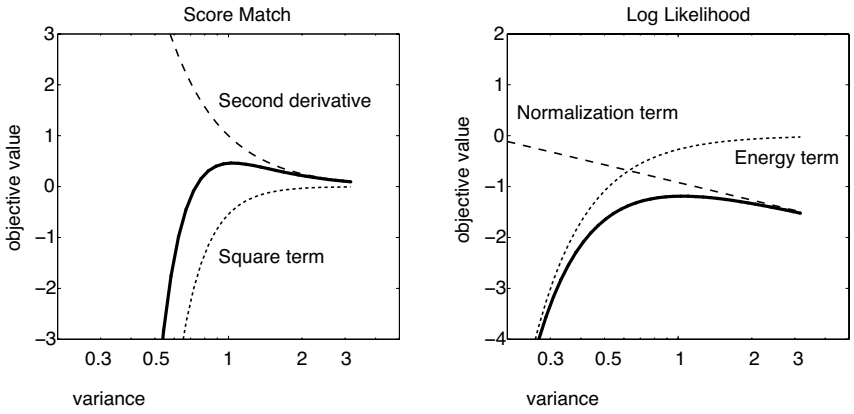

Figure 1. Illustration of score matching, applied to infer the variance of a univariate normal distribution. Both the score matching objective and the log-likelihood (solid curves) have the optimum at the correct position, but the functions differ significantly in shape. Comparing the two terms of the score matching objective with the energy and normalization term of the log-likelihood reveals the similarity between the second derivative term in the score matching objective and the normalization term: both penalize unspecific models with unnecessarily high variance.

To obtain the score matching estimate of the mean μ we take the derivative and set it to zero to obtain

$$\frac{\partial J}{\partial \mu} = \frac{1}{T} \sum_{t=1}^{T} \sigma^{-4} (\mathbf{x}(t) - \mu) = 0 \tag{16}$$

$$\hat{\mu} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{x}(t) \tag{17}$$

which is the sample mean, in agreement with the maximum likelihood estimate. Similarly, to fit the variance σ^2 we take the derivative

$$\frac{\partial J}{\partial \sigma} = \frac{1}{T} \sum_{t=1}^{T} \left(-2\sigma^{-5} (\mathbf{x}(t) - \mu)^2 \right) + 2\sigma^{-3} = 0 \tag{18}$$

$$\hat{\sigma} = \sqrt{\sum_{t=1}^{T} (\mathbf{x}(t) - \mu)^2} \tag{19}$$

which is the sample variance, again in accordance with maximum likelihood. In general, there is no closed form solution for the optimal parameters, so gradient methods have to be used for optimization.

The relation between score matching and maximum likelihood are further illustrated in Fig. 1. Comparing the terms in the score matching objective with the log-likelihood allows an intuitive interpretation of score matching. The square term in the score matching objective acts similar to the energy (i.e. the non-normalized log-likelihood) and tries to make the model general enough to cover all of the data points. The second derivative has a similar effect to a normalization term, penalizing the unspecific model and forcing it to focus probability mass to where the observed data lies.

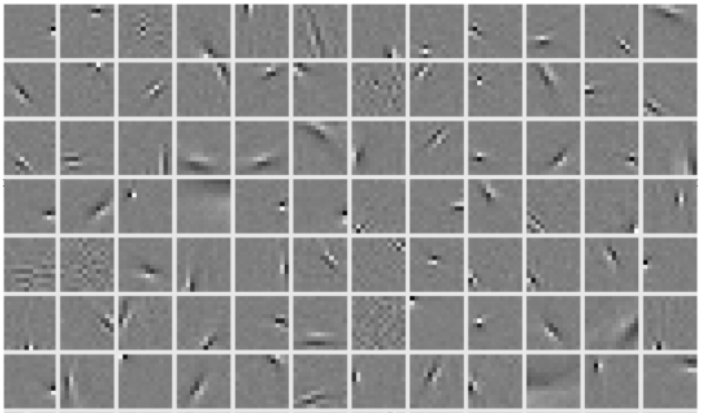

Figure 2. Overcomplete ICA model estimated with score matching. Only a subset of the 2304 filters from the 16 times overcomplete model are shown. Note that in contrast to the sparse coding model, we estimate an overcomplete set of filters rather than basis functions. This leads to some very different properties of the model.

1.2 Overcomplete ICA Example

Score matching can easily be applied to estimate ICA models, and because the normalization constant is not required to be known, it is an obvious choice for overcomplete ICA. In this case the model is

$$-\log p(\mathbf{x}) \propto E(\mathbf{x}) = \sum_{i=1}^{M} g(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}) \tag{20}$$

where the number of filters M is larger than the dimensionality of the data. Models of this kind, also known as *Products of Experts* (PoE) have been studied extensively [6], where they were estimated with *Contrastive Divergence* (CD) [7], a method that shares some similarities with score matching and is based on Monte Carlo methods.

Estimating this model for natural image patches again leads to familiar Gabor-like filters, which is shown in Fig. 2 for a 16 times overcomplete model. Here we show the filters $\mathbf{w}$ rather than basis functions, because in the overcomplete model the filter matrix cannot be inverted, and thus no basis functions are defined.

The complete ICA model which we considered in the previous section can be seen as a bridge between overcomplete sparse coding models on one hand, and overcomplete PoE or ICA models on the other hand, which have some very important differences. In an energy-based model, the internal representation can be computed by a fast, simple feed-forward computation, whereas in the generative sparse coding model, the optimal pattern of activities $\mathbf{s}$ is the solution to an optimization problem. This implicitly nonlinear mapping between the data $\mathbf{x}$ and the components $\mathbf{s}$ has some attractive properties for neuroscience: since the basis functions inhibit each other, behavior such as end-stopping and nonclassical surround effects has been observed in overcomplete sparse coding models [8]. On the other hand, a recent study [9] has shown some advantages in energy based models over generative models in denoising applications, and it is not clear at the time how the two approaches are related and which provides a better model for natural images, and therefore ultimately for visual processing.

References

[1] A. Hyvärinen. Estimation of non-normalized statistical models using score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

[2] A. Hyvärinen. Connections between score matching, contrastive divergence, and pseudolikelihood for continuous-valued variables. *IEEE Transactions on Neural Networks*, 18(5):1529–1531, 2007.

[3] A. Hyvärinen. Some extensions of score matching. *Computational Statistics and Data Analysis*, 51:2499–2512, 2007.

[4] U. Köster and A. Hyvärinen. A two-layer ICA-like model estimated by score matching. In *Artificial Neural Networks - ICANN 2007, Lecture Notes in Computer Science*, pages 798–807. Springer Berlin / Heidelberg, 2007.

[5] U. Köster, A. Hyvärinen, and J. T. Lindgren. Estimating Markov random field potentials for natural images. In *ICA 2007, Lecture Notes in Computer Science*, pages 515–522. Springer Berlin / Heidelberg, 2009.

[6] G. E. Hinton. Products of experts. In *Proceedings of the Ninth International Conference on Artificial Neural Networks (ICANN)*, volume 1, pages 1–6, 1999.

[7] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[8] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems 19*, pages 801–808. MIT Press, 2006.

[9] M. Elad, P. Milanfar, and R. Rubinstein. Analysis versus synthesis in signal priors. In *Inverse Problems 23 (2007*, pages 947–968, 2005.