Deep PDF: Density Estimation for Measurement Model Learning
Supplementary Material

Dmitry Kopitkov and Vadim Indelman

This document provides supplementary material to the paper [1]. Therefore, it should not be considered a self-contained document, but instead regarded as an appendix of [1].

Appendix A: Surface Change after Gradient Descent Update of $L(\theta, X) = f(X;\theta)$

Performing a single gradient descent step w.r.t. loss $L(\theta, X) = f(X;\theta)$ at a specific *optimized point* X results in a weights update $\theta_i = \theta_{i-1} - \delta \cdot \nabla L$, where δ is step size and $\nabla L = \frac{\partial f(x;\theta)}{\partial \theta}|_{x=X,\theta=\theta_{i-1}}$. After such update, the surface height at point X, $f(X;\theta)$, can be approximated via a Taylor expansion as:

$$\begin{aligned} f(X;\theta_i) &= f(X;\theta_{i-1}) + (\theta_i - \theta_{i-1})^T \cdot \frac{\partial f(x;\theta)}{\partial \theta}|_{x=X,\theta=\theta_{i-1}} + \\ &+ \frac{1}{2} \cdot (\theta_i - \theta_{i-1})^T \cdot H_\theta \cdot (\theta_i - \theta_{i-1}) + \cdots = \\ &= f(X;\theta_{i-1}) - \delta \cdot (\frac{\partial f(x;\theta)}{\partial \theta}|_{x=X,\theta=\theta_{i-1}})^T \cdot \frac{\partial f(x;\theta)}{\partial \theta}|_{x=X,\theta=\theta_{i-1}} + \\ &+ \frac{1}{2}\delta^2 \cdot (\frac{\partial f(x;\theta)}{\partial \theta}|_{x=X,\theta=\theta_{i-1}})^T \cdot H_\theta \cdot (\frac{\partial f(x;\theta)}{\partial \theta}|_{x=X,\theta=\theta_{i-1}}) + \cdots , \end{aligned} \quad (1)$$

where:

$$H_\theta \triangleq \frac{\partial^2 f(x;\theta)}{\partial \theta^2}|_{x=X,\theta=\theta_{i-1}}. \quad (2)$$

Above we can see the first two elements of Taylor expansion where the second element depends on a quadratic step size δ^2. The (learning) step size is typically less than one. Further, assuming it is very small number ($\delta \ll 1$), $f(X;\theta_i)$ can

D. Kopitkov is with the Technion Autonomous Systems Program (TASP), Technion - Israel Institute of Technology, Haifa 32000, Israel, dimkak@tx.technion.ac.il . V. Indelman is with the Department of Aerospace Engineering, Technion - Israel Institute of Technology, Haifa 32000, Israel, vadim.indelman@technion.ac.il.

be approximated good enough by a first-order Taylor expansion as:

$$f(X;\theta_i) = f(X;\theta_{i-1}) - \delta \cdot (\frac{\partial f(x;\theta)}{\partial \theta}|_{x=X,\theta=\theta_{i-1}})^T \cdot \frac{\partial f(x;\theta)}{\partial \theta}|_{x=X,\theta=\theta_{i-1}}, \quad (3)$$

and define

$$df(X) \triangleq f(X;\theta_i) - f(X;\theta_{i-1}) = -\delta \cdot g(X, X, \theta_{i-1}), \quad (4)$$

$$g(X_1, X_2, \theta) \triangleq \frac{\partial f(X_1;\theta)}{\partial \theta}^T \cdot \frac{\partial f(X_2;\theta)}{\partial \theta}, \quad (5)$$

where $g(\cdot)$ function calculates the *cross-gradient* similarity between two different points X_1 and X_2, i.e. the dot product between θ gradients at these two points. Further, $g(X, X, \theta_{i-1})$ is the L2 norm of θ gradient at point X.

Similarly, the change of surface height at point X', which is different from the *optimized point* X (i.e. $X' \neq X$), can be approximated as:

$$df(X') = -\delta \cdot g(X', X, \theta_{i-1}). \quad (6)$$

Note that the cross-gradient similarity function is symmetric, i.e. $g(X', X, \cdot) = g(X, X', \cdot)$. Thus, from the above we can see that for any two arbitrary points $X \in \mathbb{R}^n$ and $X' \in \mathbb{R}^n$, pushing at one point will change height at another point according to the cross-gradient similarity $g(X', X, \cdot)$. Therefore, $g(X', X, \cdot)$ is responsible for correlation in height change at different points. Given $|g(X', X, \cdot)|$ is small (or zero), optimizing loss L (pushing surface) at point X will not affect the surface at point X'. The opposite is also true: if $|g(X', X, \cdot)|$ is large, then pushing the surface at point X will have a big impact at point X'.

■

Appendix B: Surface Change after Gradient Descent Update of Simple Loss

In case of simple loss $L(\theta, X_U, X_D) = -f(X_U;\theta) + f(X_D;\theta)$ where X_U and X_D are sampled from $\mathbb{P}^U$ and $\mathbb{P}^D$ respectively, consider weights update $\theta_i = \theta_{i-1} - \delta \cdot \nabla L$. Similarly to Eq. (1), the change of surface height at some point $X \in \mathbb{R}^n$ can be approximated as a first-order Taylor expansion:

$$df(X) = \delta \cdot g(X_U, X, \theta_{i-1}) - \delta \cdot g(X_D, X, \theta_{i-1}). \quad (7)$$

Further, considering the stochastic nature of X_U and X_D, the average differential $\mathbb{E}[df(X)]$ can be calculated as:

$$\mathbb{E}[df(X)] = \delta \cdot \int \mathbb{P}^U(X') \cdot g(X', X, \theta_{i-1}) dX' -$$
$$-\delta \cdot \int \mathbb{P}^D(X') \cdot g(X', X, \theta_{i-1}) dX' = \delta \cdot \int [\mathbb{P}^U(X') - \mathbb{P}^D(X')] \cdot g(X', X, \theta_{i-1}) dX'. \quad (8)$$

Again, we see that $g(X', X, \cdot)$ has the role of a filter that controls correlations between points. In contrast, the term $[\mathbb{P}^U(X') - \mathbb{P}^D(X')]$ depends only on point X'.

■

Appendix C: Proof of Divergence of Simple Loss

In case of simple loss $L(\theta, X_U, X_D) = -f(X_U;\theta) + f(X_D;\theta)$ the average differential $\mathbb{E}[df(X)]$,

$$\mathbb{E}[df(X)] = \delta \cdot \int [\mathbb{P}^U(X') - \mathbb{P}^D(X')] \cdot g(X', X, \theta_{i-1}) dX', \tag{9}$$

contains *signal* term $[\mathbb{P}^U(X') - \mathbb{P}^D(X')]$ and *filter* term $g(X', X, \theta_{i-1})$. Since the *signal* term does not depend on θ and is unchanging along the optimization, there is no stable point for the optimization convergence. Thus, the learning process diverges. This can be proved by contradiction as follows.

Assume, in contradiction, that the above simple loss converges at optimization step t to the loss's local minimum. Then, for the following optimization steps, θ, and so $g(X', X, \theta)$, will remain (almost) constant - GD will stay in the minimum area. Hence, the integral in Eq. (9) becomes a constant for each given point X, since θ does not change anymore. This constant is non-zero, unless both *up* and *down* densities are identical $\mathbb{P}^U(X') = \mathbb{P}^D(X')$; or the learned $f(X;\theta)$ is flat ($g(X', X, \theta) = 0$ for any two points). Yet, such specific cases are degenerate and we exclude them.

Concluding from the above, after optimization step t the $\mathbb{E}[df(X)]$ became non-zero constant for each X. Therefore, on average the surface height continues to change by a non-zero constant at each point X, going towards $\pm\infty$. Yet, such change of surface $f(X;\theta)$ is impossible since θ is (almost) constant after t. We got a contradiction, meaning that the above convergence assumption is wrong; that is, the simple loss $L(\theta, X_U, X_D) = -f(X_U;\theta) + f(X_D;\theta)$ cannot converge due to non-zero signal $[\mathbb{P}^U(X') - \mathbb{P}^D(X')]$.

Appendix D: Surface Change after Gradient Descent Update of PDF Loss

In case of *pdf loss* $L_{pdf}(\theta, X_U, X_D) =$
$-f(X_U;\theta) \cdot \mathbb{P}^D(X_U) + f(X_D;\theta) \cdot \Big[f(X_D;\theta)\Big]^{mf}$ where X_U and X_D are sampled from $\mathbb{P}^U$ and $\mathbb{P}^D$ respectively, consider weights update $\theta_i = \theta_{i-1} - \delta \cdot \nabla L$. Similarly to Eq. (1), the change of surface height at some point $X \in \mathbb{R}^n$ can be approximated as first-order Taylor expansion:

$$df(X) = \delta \cdot \mathbb{P}^D(X_U) \cdot g(X_U, X, \theta_{i-1}) - \delta \cdot f(X_D;\theta) \cdot g(X_D, X, \theta_{i-1}). \tag{10}$$

Further, considering stochastic nature of X_U and X_D, the average differential $\mathbb{E}[df(X)]$ can be calculated as:

$$\begin{aligned}\mathbb{E}[df(X)] = \delta \cdot \int \mathbb{P}^U(X') \cdot \mathbb{P}^D(X') \cdot g(X', X, \theta_{i-1}) dX' - \\ - \delta \cdot \int \mathbb{P}^D(X') \cdot f(X'; \theta) \cdot g(X', X, \theta_{i-1}) dX' = \\ = \delta \cdot \int \mathbb{P}^D(X') \cdot [\mathbb{P}^U(X') - f(X'; \theta)] \cdot g(X', X, \theta_{i-1}) dX'. \quad (11)\end{aligned}$$

■

Appendix E: PDF Loss as Empirical Approximation of Continuous Squared Loss

Consider a squared loss between NN surface $f(X;\theta)$ and a target function $\mathbb{P}^U(X)$, weighted by the pdf $\mathbb{P}^D$ of *down* distribution:

$$L(\theta) = \mathbb{E}_{X\sim\mathbb{P}^D}\left[\left[\mathbb{P}^U(X) - f(X;\theta)\right]^2\right] = \int \mathbb{P}^D(X) \cdot \left[\mathbb{P}^U(X) - f(X;\theta)\right]^2 dX. \quad (12)$$

Similarly to the ratio estimation methods in [2], we can show that the above loss can be minimized by minimizing our *pdf loss*, as follows:

$$L(\theta) = \int \mathbb{P}^D(X) \cdot \left[\mathbb{P}^U(X)\right]^2 + \mathbb{P}^D(X) \cdot \left[f(X;\theta)\right]^2 - 2 \cdot \mathbb{P}^D(X) \cdot \mathbb{P}^U(X) \cdot f(X;\theta) dX. \quad (13)$$

Since the first term doesn't depend on θ, we can rewrite the loss as:

$$\begin{aligned}L(\theta) = \int \mathbb{P}^D(X) \cdot \left[f(X;\theta)\right]^2 - 2 \cdot \mathbb{P}^D(X) \cdot \mathbb{P}^U(X) \cdot f(X;\theta) dX = \\ = \int \mathbb{P}^D(X) \cdot \left[f(X;\theta)\right]^2 dX - 2 \cdot \int \mathbb{P}^U(X) \cdot \mathbb{P}^D(X) \cdot f(X;\theta) dX = \\ = \mathbb{E}_{X\sim\mathbb{P}^D}[f(X;\theta)^2] - 2 \cdot \mathbb{E}_{X\sim\mathbb{P}^U}[\mathbb{P}^D(X) \cdot f(X;\theta)]. \quad (14)\end{aligned}$$

The above can be approximated by N samples $\{X_U^i\}_{i=1}^N$ from distribution $\mathbb{P}^U$ and N samples $\{X_D^i\}_{i=1}^N$ from distribution $\mathbb{P}^D$ as:

$$L(\theta) = \frac{1}{N}\sum_{i=1}^N f(X_D^i;\theta)^2 - 2 \cdot \frac{1}{N}\sum_{i=1}^N \mathbb{P}^D(X_U^i) \cdot f(X_U^i;\theta) \quad (15)$$

where for notation simplicity we can rewrite it as one-sample version:

$$L(\theta) = f(X_D;\theta)^2 - 2 \cdot \mathbb{P}^D(X_U) \cdot f(X_U;\theta). \quad (16)$$

Next, note that the above loss has the same gradient w.r.t. θ as the following loss:

$$L(\theta) = 2 \cdot f(X_D;\theta) \cdot \Big[f(X_D;\theta)\Big]^{mf} - 2 \cdot \mathbb{P}^D(X_U) \cdot f(X_U;\theta), \tag{17}$$

where $\Big[\cdot\Big]^{mf}$ specifies *magnitude function* terms that only produce magnitude coefficients but whose gradient w.r.t. θ is not calculated (*stop gradient* in Tensorflow [3]). Thus, both losses are equal in context of the gradient-based optimization.

After changing order of terms in Eq. (17), and deviding by 2 (e.g. moving it into learning rate), we will finally get the *pdf loss*:

$$L_{pdf}(\theta, X_U, X_D) = -f(X_U;\theta) \cdot \mathbb{P}^D(X_U) + f(X_D;\theta) \cdot \Big[f(X_D;\theta)\Big]^{mf}. \tag{18}$$

■

Appendix F: DeepPDF algorithm in more detail

Inputs:
G^U : generator of data from *up* distribution $\mathbb{P}^U$ (e.g. dataset sampled from this distribution)
G^D : generator of data from *down* distribution $\mathbb{P}^D$ (e.g. Normal or Uniform sampler)
N : batch size
$f(X;\theta)$: neural network whose input has dimension of data, and whose output is a scalar
Outputs: $f(X;\theta)$: neural network approximation of target pdf $\mathbb{P}^U(X)$
begin:
  while *not converged* **do**
    $\{X_U^i\}_{i=1}^N \Longleftarrow$ sample N points from G^U
    $\{X_D^i\}_{i=1}^N \Longleftarrow$ sample N points from G^D
    $L_T(\theta) = \frac{1}{N}\sum_{i=1}^N L_{pdf}(\theta, X_U^i, X_D^i)$
    Perform one optimization update on θ to minimize: $\min_\theta L_T(\theta)$ (e.g. Gradient Descent, Adam, BFGS, etc.)
  end
end

Algorithm 1: `DeepPDF` learns pdf $\mathbb{P}^U(X)$ using only samples from distribution $\mathbb{P}^U$.

Appendix G: Point optimization experiment

In this experiment we apply our loss only at one specific point $X_z = [0,0]^T$ in $\mathbb{R}^2$ space, to demonstrate that the described PSO works in a simple case of

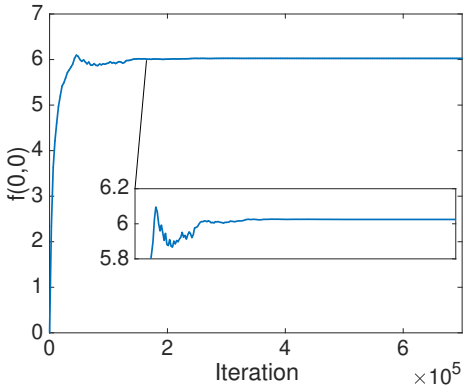

Figure 1: Surface height $f(X_z;\theta)$ at $X_z = [0,0]^T$ during optimization with point loss $L_{zero}(\theta)$ in Eq. (19).

single point optimization. The used loss is:

$$L_{zero}(\theta) = -f(X_z;\theta)\cdot\mathbb{1}_U[X = X_z] + f(X_z;\theta)\cdot\Big[f(X_z;\theta)\Big]^{mf}\cdot\mathbb{1}_D[X = X_z], \quad (19)$$

where $\mathbb{1}_U[X = X_z]$ is indicator random variable that is equal to 1 with probability $p_U = 0.9$ and is 0 with probability $1 - p_U$. Similarly, $\mathbb{1}_D[X = X_z]$ is 1 with probability $p_D = 0.15$ and is 0 otherwise. In such a way we simulate a sampling process from $\mathbb{P}^U(X)$ and $\mathbb{P}^D(X)$ where we assume $\mathbb{P}^U(X_z) = p_U$ and $\mathbb{P}^D(X_z) = p_D$. According to the analysis from Section [3] , the balance between up/down forces using $L_{zero}(\theta)$ should be at:

$$F_U(X_z) = F_D(X_z) \quad \Longrightarrow \quad p_U = p_D \cdot f(X_z;\theta) \quad \Longrightarrow \quad f(X_z;\theta) = \frac{p_U}{p_D} = 6. \quad (20)$$

We apply Adam optimizer [4] with loss $L_{zero}(\theta)$ during several thousand iterations and show value of $f(X_z;\theta)$ along this optimization in Figure 1. As can be seen, $f(X_z;\theta)$ quickly reaches height 6, following stochastic oscillation around this value. With learning rate decay the oscillations become smaller and the final $f(X_z;\theta)$ value ends to be very close to 6. This simple experiment demonstrates the power of PSO to approximate height of a target pdf. Additionally, it provides insight on how PSO can be used to infer density of discrete random variable instead of continuous, which was the main focus of this paper.

Appendix H: Used Densities - Equations

Below are the analytical density functions for distributions used in our experiments.

1. *Cosine:*

$$\mathbb{P}^U(x_1, x_2) = \frac{1}{\mu} \cdot \big[\cos(4x_1 \cdot x_2) + 1\big], \quad \mu = 17.631302268269998 \tag{21}$$

2. *Columns:*

$$\mathbb{P}^U(x_1, x_2) = p(x_1) \cdot p(x_2), \tag{22}$$

where $p(\cdot)$ is mixture distribution with 5 components:
$Uniform(-2.3, -1.7), \mathcal{N}(-1.0, std = 0.2), \mathcal{N}(0.0, std = 0.2), \mathcal{N}(1.0, std = 0.2), Uniform(1.7, 2.3)$
Each component has weight 0.2.

3. *RangeMsr:*

$$\mathbb{P}^U(x, y, f) = p(x) \cdot p(y) \cdot p(f|x, y), \tag{23}$$

where

$$p(x) = p(y) = Uniform(-2.0, 2.0), \tag{24}$$

and

$$p(f|x, y) = \mathcal{N}(\sqrt{x^2 + y^2}, std = 1.0). \tag{25}$$

References

[1] D. Kopitkov and V. Indelman, "Deep pdf: Density estimation for measurement model learning," *IEEE Intl. Conf. on Robotics and Automation (ICRA) and IEEE Robotics and Automation Letters (RA-L), Mutual Submission*, May 2019.

[2] M. Sugiyama, T. Suzuki, and T. Kanamori, *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

[3] "TensorFlow," www.tensorflow.org. [Online]. Available: www.tensorflow.org

[4] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.