A Proposal for Supervised Density Estimation

Robert A. Vandermeulen
Machine Learning Group
Technische Universität Berlin
vandermeulen@tu-berlin.de

René Saitenmacher
Machine Learning Group
Technische Universität Berlin
r.saitenmacher@tu-berlin.de

Alexander Ritchie
Department of EECS
University of Michigan
aritch@umich.edu

Abstract

Density estimation is the unsupervised task of estimating an unknown probability density given samples from the density. In this paper we consider a supervised density estimation setting where, in addition to samples from the target density, one has access to an auxiliary collection of samples which are known to not be well-represented according to the target density. We derive two theoretically principled, highly flexible, and computationally efficient approaches to the supervised density estimation setting which incorporate the auxiliary samples to improve the density estimation task. To ascertain the overall usefulness of the supervised density estimation setting, and our approaches in particular, we propose a few experiments on models ranging from classic Gaussian mixture models to more recent variational autoencoders.

1 Introduction

Perhaps one of the most basic problems in statistics is to estimate an unknown probability density p given samples $X_1, \dots, X_n \sim p$. This task is also quite common in machine learning where it fits squarely into the "unsupervised" learning paradigm. Many machine learning settings have been proposed that blur the lines between more classic settings including "semi-supervised" learning [32, 31, 3], "weakly-supervised" learning [30], and supervised learning approaches to unsupervised tasks [22]. A somewhat paradoxical question, which we propose here, is how to construct, or whether it is even useful to investigate, a supervised approach to density estimation. Specifically we consider the setting where one has access to two collections of data in $\mathbb{R}^d$, one of which $X_1, \dots, X_n \sim p$ is distributed according to a *target* density p, and another collection $\widetilde{X}_1, \dots, \widetilde{X}_m \sim q$ is distributed according to a *contrast* density that is representative of how we do *not* want our density to look.

As a motivating example for supervised density estimation we consider the use of a density estimator for anomaly detection where low likelihood examples according to the estimated density are deemed anomalous [23]. Despite typically being an unsupervised task trained on a collection of nominal data, it has been observed that incorporating known anomalies into the unsupervised learning objective is helpful for anomaly detection with density level set estimation [8, 24], even when the anomalous training samples are not representative of anomalies seen during test time. Though [8, 24] are not density estimation methods, it is natural to suspect that the inclusion of anomalous samples may produce better density estimates. In the next section we propose two principled, highly flexible, and computationally efficient methods for incorporating supervision into density estimation, followed by a discussion of related works in Section 3. In Section 4 we propose experiments to explore the behavior of supervised density estimation and determine its value for other use cases.

Pre-registration workshop NeurIPS (2020), Vancouver, Canada.

2 Proposed Losses

In this section we propose two methods for adding supervision to a density estimation objective. For a class of distributions $\mathcal{F}$ we denote the general density estimation problem as $\arg\min_{\widehat{p}\in\mathcal{F}} R(\widehat{p})$, where R is some objective function utilizing the samples $X_1, \ldots, X_n \sim p$. For maximum likelihood estimation, for example, we have that $R(f) = \frac{1}{n}\sum_{i=1}^{n} -\log\left(f\left(X_i\right)\right)$. For the supervised density estimation setting we propose an objective of the form

$$\arg\min_{\widehat{p}\in\mathcal{F}} R(\widehat{p}) + \lambda V(\widehat{p}) \tag{1}$$

where $\lambda > 0$ and V is a term which penalizes $\widehat{p}$ in some way so as to incorporate supervision into the objective utilizing samples from a contrast distribution. To help motivate the proposed penalty terms we will derive population versions which utilize the exact contrast density q followed by finite sample versions which use the contrast samples $\widetilde{X}_1, \ldots, \widetilde{X}_m \sim q$.

2.1 Coding Theory Objective

For our first objective we consider $\widehat{p}$ from a coding theory perspective. We would like to enforce that using a coding scheme optimized for $\widehat{p}$ produces a coding scheme that is more optimal for coding p than for coding q. This notion can be encapsulated in the following objective, where $\varepsilon > 0$,

$$\arg\min_{\widehat{p}\in\mathcal{F}} R\left(\widehat{p}\right) \text{ such that } D_{KL}\left(q\|\widehat{p}\right) - D_{KL}\left(p\|\widehat{p}\right) \geq \varepsilon. \tag{2}$$

The inequality constraint in (2) is equivalent to

$$\int q(x)\left(\log q(x) - \log\widehat{p}(x)\right)dx - \int p(x)\left(\log p(x) - \log\widehat{p}(x)\right)dx \geq \varepsilon.$$

Incorporating the constant terms into ε' and multiplying by -1 we get that this is equivalent to

$$\int q(x)\log\widehat{p}(x)dx - \int p(x)\log\widehat{p}(x)dx \leq -\varepsilon'. \tag{3}$$

The left hand side of (3) is not bounded from below, however in [11] it was suggested to use clipping to resolve this, yielding our first supervised form

$$\widehat{V}_{base}\left(f\right) = \frac{1}{n}\sum_{i=1}^{n}\max\left\{0, D + \log\widehat{p}\left(\widetilde{X}_i\right) - \log\widehat{p}\left(X_i\right)\right\} \tag{4}$$

where D is the number of pixels for image data and we will let $d = D$ for non-image data. We will use (4) as a baseline loss. Our approach is to instead use the Lagrangian form of (2), yielding an unconstrained optimization problem in the form of (1). We write an equivalent constraint to (3) as

$$\exp\left\{\int q(x)\log\widehat{p}(x)dx - \int p(x)\log\widehat{p}(x)dx\right\} \leq \delta,$$

for $\delta = \exp\left(-\varepsilon'\right)$. Because the left hand side of the previous inequality is positive we can restate the original objective (2) in its Lagrangian form

$$\arg\min_{\widehat{p}\in\mathcal{F}} R\left(\widehat{p}\right) + \lambda\exp\left\{\int q(x)\log\widehat{p}(x)dx - \int p(x)\log\widehat{p}(x)dx\right\} \tag{5}$$

where $\lambda > 0$. Rewriting the right term in the last line in terms of expectations we have

$$\exp\left\{\mathbb{E}_{\widetilde{X}\sim q}[\log\widehat{p}(\widetilde{X})] - \mathbb{E}_{X\sim p}\left[\log\widehat{p}\left(X\right)\right]\right\}$$

which gives us our next supervised term

$$\widehat{V}_{code}(f) = \exp\left\{\frac{1}{m}\sum_{i=1}^{m}\log\widehat{p}(\widetilde{X}_i) - \frac{1}{n}\sum_{j=1}^{n}\log\widehat{p}\left(X_j\right)\right\}. \tag{6}$$

We will now find an upper bound for the right term in (5) that has the form of a sample mean thereby making it amenable to stochastic gradient descent (SGD) optimization for use with deep methods. We have that

$$\begin{aligned}
&\exp\left\{\int q(x)\log\widehat{p}(x)dx - \int p(x)\log\widehat{p}(x)dx\right\} \\
&= \exp\left\{\int q(x)\log\widehat{p}(x)dx\right\}\exp\left\{-\int p(x)\log\widehat{p}(x)dx\right\} \\
&= \exp\left\{\mathbb{E}_{\widetilde{X}\sim q}[\log\widehat{p}(\widetilde{X})]\right\}\exp\left\{\mathbb{E}_{X\sim p}\left[-\log\widehat{p}(X)\right]\right\} \\
&\leq \mathbb{E}_{\widetilde{X}\sim q}[\exp\{\log\widehat{p}(\widetilde{X})\}]\mathbb{E}_{X\sim p}\left[\exp\left\{-\log\widehat{p}(X)\right\}\right] \qquad (7) \\
&= \mathbb{E}_{\widetilde{X}\sim q}\left[\widehat{p}(\widetilde{X})\right]\mathbb{E}_{X\sim p}\left[\left(\widehat{p}(X)\right)^{-1}\right] = \mathbb{E}_{(X,\widetilde{X})\sim p\times q}\left[\widehat{p}(\widetilde{X})/\widehat{p}(X)\right] \qquad (8)
\end{aligned}$$

where (7) follows from Jensen's Inequality and the fact that the two factors are non-negative and (8) follows from statistical independence. If $m = n$ then we have the following finite sample version of the last expression which can be used in SGD, with a small $\eta > 0$ included for stability:

$$\widehat{V}_{codeSGD}\left(\widehat{p}\right) = \sum_{i=1}^{n}\frac{\widehat{p}(\widetilde{X}_i)}{\widehat{p}(X_i)+\eta}. \qquad (9)$$

2.2 Classification Loss

The following loss encourages our density estimate $\widehat{p}$ to be such that there exists a classifier that can easily differentiate between samples from $\widehat{p}$ and q. For densities f and g let $\beta(f,g)$ be the Bayes risk for the classification problem of differentiating between f and g with prior class probabilities $1/2$. The following is the optimization problem we would like to solve

$$\arg\min_{\widehat{p}\in\mathcal{F}} R\left(\widehat{p}\right) + \lambda\beta\left(\widehat{p},q\right). \qquad (10)$$

From [21] (see (8) in their work) we have that $\beta(f,g) \leq \frac{1}{2}\int\sqrt{f(x)}\sqrt{g(x)}dx$. Note for $x = t$ we have that the tangent line at $(x, y = \sqrt{x})$ is given by $y - \sqrt{t} = \frac{1}{2}t^{-1/2}(x-t)$. Since the square root function is concave we have for all $t > 0$ and $x \geq 0$ that $\sqrt{x} \leq \frac{1}{2}\left(x/\sqrt{t} + \sqrt{t}\right)$. Now we have the following, where γ is some density that is positive on the support of $\widehat{p}$ which is used to get a Monte Carlo estimate of $\int\sqrt{\widehat{p}(x)}dx$

$$\begin{aligned}
\beta\left(f,g\right) &\leq \frac{1}{2}\int\sqrt{\widehat{p}(x)}\sqrt{q(x)}dx \leq \frac{1}{2}\min_{t>0}\int\sqrt{\widehat{p}(x)}\frac{1}{2}\left(q(x)/\sqrt{t}+\sqrt{t}\right) \\
&= \frac{1}{4}\left(\min_{t>0}\int\sqrt{\widehat{p}(x)}\left(q(x)/\sqrt{t}\right)dx + \int_{\{x|\gamma(x)>0\}}\frac{\gamma(x)}{\gamma(x)}\sqrt{\widehat{p}(x)}\sqrt{t}dx\right) \\
&= \frac{1}{4}\left(\min_{t>0}\frac{1}{\sqrt{t}}\underbrace{\mathbb{E}_{\widetilde{X}\sim q}\left[\sqrt{\widehat{p}(\widetilde{X})}\right]}_{:=E_1} + \sqrt{t}\underbrace{\mathbb{E}_{U\sim\gamma}\left[\gamma(U)^{-1}\sqrt{\widehat{p}(U)}\right]}_{:=E_2}\right) =: \min_{t>0}h(t). \qquad (11)
\end{aligned}$$

Setting the derivative of $h(t)$ equal to zero, we obtain the minimizing $t > 0$

$$h'(t) = \frac{1}{4}\left(-\frac{1}{2}\frac{1}{\sqrt{t^3}}E_1 + \frac{1}{2}\frac{1}{\sqrt{t}}E_2\right) = \frac{1}{8\sqrt{t^3}}\left(tE_2 - E_1\right) = 0 \implies t^* = \frac{E_1}{E_2}.$$

Observe that $t^* > 0$ as both expectations are positive. Thus, the value at the minimum is given by

$$\begin{aligned}
h(t^*) &= \frac{1}{4}\left(\sqrt{\frac{E_2}{E_1}}E_1 + \sqrt{\frac{E_1}{E_2}}E_2\right) \\
&= \frac{2}{4}\sqrt{E_1E_2} = \frac{1}{2}\sqrt{\mathbb{E}_{(U,\widetilde{X})\sim\gamma\times q}\left[\gamma(U)^{-1}\sqrt{\widehat{p}(U)}\sqrt{\widehat{p}(\widetilde{X})}\right]} \qquad (12)
\end{aligned}$$

where (12) follows from statistical independence. For $U_1, \ldots, U_m \overset{iid}{\sim} \gamma$ we get the following finite sample version of (12) which is our last supervised term

$$\widehat{V}_{class}\left(\widehat{p}, q\right) = \frac{1}{2}\sqrt{\frac{1}{m}\sum_{i=1}^{m}\frac{\sqrt{\widehat{p}(U_i)\widehat{p}(\widetilde{X}_i)}}{\gamma(U_i)}}. \tag{13}$$

To make this amenable to SGD we will simply square it, which is a strictly monotonic mapping since (13) is always nonnegative

$$\widehat{V}_{classSGD}\left(\widehat{p}, q\right) = \frac{1}{4m}\sum_{i=1}^{m}\frac{\sqrt{\widehat{p}(U_i)\widehat{p}(\widetilde{X}_i)}}{\gamma(U_i)}. \tag{14}$$

3 Related Work

The loss in (4) was introduced in [11] as a way to improve anomaly detection utilizing language and neural autoregressive models. In that paper they propose samples from large, easily available datasets to be used as anomalies during training, a technique they term *outlier exposure*. Works like [13] attempt to unify density estimators with positive and negative examples but do not yield a single density estimator incorporating both classes of samples. The following are a few settings and techniques which appear similar to our setting, but are in fact different.

Ranking losses (e.g., contrastive loss [5], triplet loss [4], or angular loss [28]) are used in metric learning to encourage (dis)similarity among data pairs or triplets in the learned metric. While not directly utilizable for generative models, these losses bear some similarity to the setting we consider in this paper. In particular, training with ranking losses encourages learned representations of similar examples to be close according to some metric (generally, Euclidean), and the opposite for dissimilar examples. In contrast, our proposed losses do not penalize representations directly, but rather the associated likelihoods or Bayes risk. In this sense, our proposed losses can be considered a generalization of ranking losses where (dis)similarity is encouraged with respect to likelihood or Bayes risk, rather than a metric.

Learning from Positive and Unlabeled Examples (LPUE) [16, 15, 1] addresses the setting where the learner has access to labeled data from a single positive class distributed according to p and a collection of unlabeled data which contains examples from both the positive class and some other class distributed according to q, with q not necessarily unlike p. Both the dataset assumptions and the learning objective (identifying q or samples coming from q) are unlike the supervised density estimation setting we present.

In **Noise Contrastive Estimation** (NCE) [10, 20] one chooses a tractable noise distribution f. An unnormalized density model can then be obtained by training a classifier that discriminates between training data $X_1, \ldots, X_n$ and samples $Y_1, \ldots, Y_m \sim f$. That is, the noise distribution f is a known, auxiliary distribution whose primary purpose is to enable fitting of a model and is not necessarily unlike p. NCE is not meant to improve learning, as in our case, by providing additional negative information during model training. In addition, our approach does not require knowing the contrast distribution q.

Negative Sampling [19] was proposed in the context of natural language processing as an adaptation of NCE that only needs samples from the noise distribution f, but does not require evaluating it. Like NCE it learns an unnormalized density model by training a classifier to discriminate between training data and samples from the noise distribution. However, it modifies the learning objective of NCE, regarding f as a constant. While Negative Sampling has been reported to work well in practice, this "hack" means that it loses the asymptotic consistency guarantees of NCE.

4 Experiments

Here we propose experiments to determine the usefulness and analyze the behavior of a supervised approach to density estimation. For all of our experiments we will include results for the unsupervised baseline $\lambda = 0$, the supervised baseline (4), and our proposed penalty terms (6) or (9) and (13) or (14), depending on the need to use SGD. As before $X_1, \ldots, X_n$ are samples from our target distribution p and $\widetilde{X}_1, \ldots, \widetilde{X}_m$ are samples coming from a contrast distribution q. We will let γ

from the classification loss be a multivariate t-distribution with 3 degrees of freedom with mean and covariance set to match the mean and covariance of the target samples for the low-dimensional experiments. To avoid estimating a high dimensional covariance matrix for the image dataset we will set the covariance to be the identity times the variance of a randomly selected entry of 100 randomly selected image samples. For all the experiments we will include results for 10 values of λ over a logarithmic scale from $\lambda = 0$ to a λ which demonstrates the limiting behavior as $\lambda \to \infty$. We leave a bit of flexibility in our experiments and will select the maximum value of λ by hand since we do not yet know the appropriate scale for λ.

4.1 Models

To allow for a comprehensive evaluation of our approach we will perform experiments with a variety of density estimation methods which we briefly list here.

Variational Autoencoder (VAE): We will use the "plain VAE" architecture described in Section 4 and Table 1 of [14] which is a standard convolutional VAE architecture with a 4-layer encoder and 5-layer decoder, accepting inputs of size $3 \times 64 \times 64$. We will optimize this model via SGD on the ELBO loss (R in our formulation) as given in Equation (2)-(4) of [14]. Following [14] we will use the RMSProp optimizer, a learning rate of 0.0003, a batch size of 64 and train until convergence. Since we are not sure how our loss terms affect optimization behavior we may adapt these parameters somewhat for our experiments. Note that the penalty term introduced by our SGD coding theory loss (9) relies on a probability ratio which might not be well preserved when one considers instead the ratio of ELBOs of the two probabilities. For this reason, we will resort to multi-sample Monte Carlo methods to obtain a better approximation for this penalty term [2].
Kernel Density Estimator (KDE): We consider the space of weighted KDEs with kernel centers coming from the target dataset using a Gaussian kernel, i.e. the space

$$\left\{ \sum_{i=1}^{n} w_i k(X_i - \cdot) \mid \sum_{i=1}^{n} w_i = 1 \text{ and } w_i \geq 0 \forall i \right\}.$$

For our R term we use the following objective mentioned in [12, 27], $R(f) = \frac{1}{n} \sum_{i=1}^{n} \|f - k(X_i - \cdot)\|_2^2$ whose minimizer is the standard KDE. The bandwidth parameter, which is also used in R, will be chosen via maximum likelihood leave one out cross validation (more details on this are included in Section 4.2). We will optimize the weighting vector w using projected gradient descent on the supervised objective.
Gaussian Mixture Model (GMM): Models will be optimized using expectation maximization with the V term included in the maximization step. The maximization step will be performed using gradient descent. The number of components will be selected using the standard approach of minimizing the Bayesian Information Criterion, $k \log n - 2\ell$, where k is the number of parameters of a model under consideration, n the number of samples available for fitting, and ℓ is the log-likelihood of the model under consideration [25].

4.2 Experimental Scenarios

Correcting Blurry Images in VAEs: It has often been noted in the literature that VAE models tend to produce less visually appealing samples compared to other types of deep generative models such as GANs or autoregressive models. In particular, samples from VAEs often appear blurry. Explanations and approaches to solve this issue have primarily focused on model architecture and expressiveness [29, 18, 26, 6, 9] as well as the similarity metric used in image space [14, 7]. We investigate whether the undesirable generation of blurry images from VAEs can be corrected through supervised density estimation. For this, we train a VAE on the Celeb-A dataset [17] downsampled to 64×64 (the target distribution). As contrast dataset we will use $\widetilde{X}_1, \ldots, \widetilde{X}_m$ comprising training samples to which randomized amounts of Gaussian blur have been applied. We will choose the parameters of the blur kernel such that the contrast samples exhibit a visually similar amount of blur as samples from the unsupervised VAE baseline. We will include random samples from the blurry contrast distribution and the unsupervised VAE baseline for visual comparison. Randomly generated images for the various λ parameters will be presented.

General Exploration: Here we will explore the effect of supervised density estimation by constructing estimators for three simple, two-dimensional, classification datasets. For a classification dataset let $\mathcal{X} = X_1, \ldots, X_n \sim p_X$ and $\mathcal{Y} = Y_1, \ldots, Y_n \sim p_Y$ be the training samples from the two classes. Both classes will be estimated in a supervised way using the same model class, either a KDE or GMM, thus yielding a pair of models $\widehat{p}_X$ and $\widehat{p}_Y$. We will perform experiments using both KDEs and GMMs since their R losses are quite different and may yield appreciably different behaviors for the supervised density estimation setting. We now describe how we will construct the estimator $\widehat{p}_X$, with $\widehat{p}_Y$ being constructed analogously. To perform hyperparameter selection for $\widehat{p}_X$ we first construct an estimator using the same model class, $\widehat{g}_X$, using only $\mathcal{X}$, so there is no supervised aspect and $\widehat{g}_X$ is just the vanilla KDE or GMM. The model hyperparameters for $\widehat{g}_X$ (bandwidth or component number) are selected using the respective criterion described in Section 4.1 and will be used for the supervised density estimator $\widehat{p}_X$. The estimator $\widehat{g}_X$ is only used for parameter selection so it is discarded. Using the previously selected model hyperparameters $\widehat{p}_X$ will be fit using $\mathcal{X}$ as target samples and $\mathcal{Y}$ as contrast examples. For the various values of λ we will report selected pseudocolor plots of the estimators to demonstrate the effect of the supervised terms as λ varies. Using the testing data we will additionally report Monte Carlo estimates of the KL-divergence and L^2 distance between the supervised density estimators $\widehat{p}_X, \widehat{p}_Y$ and their respective target densities p_X and p_Y. For model fitting we will use 100 samples for each class. We use three two-dimensional datasets to investigate different amounts of overlap: two moons (no overlap), Banana (some overlap), and two two-dimensional Gaussian distributions $\mathcal{N}(\mathbf{0}, I)$ and $\mathcal{N}\left(\left[0, \frac{1}{2}\right], I\right)$ (significant overlap).

Enforcing Constraints Enforcing distributional constraints in a density estimator can be a difficult task. For the supervised density estimation setting one can simply provide examples of constraint violation through $\widetilde{X}_1, \ldots, \widetilde{X}_m$ instead of analytically integrating the constraints into the model. For this experiment we will use the KDE and GMM with p being the two dimensional multivariate Gaussian $\mathcal{N}(\mathbf{0}, I)$ truncated to the unit ball (reject samples lying outside the unit ball) with $n = 50$ and these samples also being used for hyperparameter selection. We set q to be the uniform distribution on $[-2, 2] \times [-2, 2]$, rejecting samples lying inside the unit ball, with $m = 200$. The model hyperparameters will be set using the same method as the "General Exploration" experiment using the target dataset. We will report the KL-Divergence and L^2 distance between p. Selected heatmaps of $\widehat{p}$ will also be included.

Acknowledgments and Disclosure of Funding

RV and RS acknowledge support by the Berlin Institute for the Foundations of Learning and Data (BIFOLD) sponsored by the German Federal Ministry of Education and Research (BMBF). RS was also supported by the EMPIR project MedalCare (ref. 18HLT07) co-funded by the European Union's Horizon 2020 research and innovation programme and the Participating States. AR was supported by departmental fellowship from University of Michigan EECS.

The authors also thank Lukas Ruff for helpful discussions relating to this work.

References

[1] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Semi-supervised novelty detection. *Journal of Machine Learning Research*, 11(99):2973–3009, 2010.

[2] Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. Density estimation using real nvp. In *International Conference on Learning Representations*, 2016.

[3] Olivier Chapelle, Bernhard Schlkopf, and Alexander Zien. *Semi-Supervised Learning*. The MIT Press, 1st edition, 2010.

[4] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(3), 2010.

[5] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

[6] Bin Dai and David Wipf. Diagnosing and enhancing vae models. *arXiv preprint arXiv:1903.05789*, 2019.

[7] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in neural information processing systems*, pages 658–666, 2016.

[8] Nico Görnitz, Marius Kloft, Konrad Rieck, and Ulf Brefeld. Toward supervised anomaly detection. *J. Artif. Int. Res.*, 46(1):235–262, January 2013.

[9] Ishaan Gulrajani, Kundan Kumar, Faruk Ahmed, Adrien Ali Taiga, Francesco Visin, David Vazquez, and Aaron Courville. Pixelvae: A latent variable model for natural images. In *International Conference on Learning Representations*, 2017.

[10] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 297–304, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. JMLR Workshop and Conference Proceedings.

[11] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019.

[12] J. Kim and C. Scott. Robust kernel density estimation. *J. Machine Learning Res.*, 13:2529–2565, 2012.

[13] M. Kristan, D. Skočaj, and A. Leonardis. Online kernel density estimation for interactive learning. *Image and Vision Computing*, 28(7):1106 – 1116, 2010. Online pattern recognition and machine learning techniques for computer-vision: Theory and applications.

[14] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, pages 1558–1566. PMLR, 2016.

[15] Xiao-Li Li and Bing Liu. Learning from positive and unlabeled examples with different data distributions. In *European conference on machine learning*, pages 218–229. Springer, 2005.

[16] Bing Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and Philip S Yu. Building text classifiers using positive and unlabeled examples. In *Third IEEE International Conference on Data Mining*, pages 179–186. IEEE, 2003.

[17] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[18] Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. Biva: A very deep hierarchy of latent variables for generative modeling. In *Advances in neural information processing systems*, pages 6551–6562, 2019.

[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA, 2013. Curran Associates Inc.

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

[21] Kevin Moon and Alfred Hero. Multivariate f-divergence estimation with confidence. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2420–2428. Curran Associates, Inc., 2014.

[22] A. Ritchie, C. Scott, L. Balzano, D. Kessler, and C. S. Sripada. Supervised principal component analysis via manifold optimization. In *2019 IEEE Data Science Workshop (DSW)*, pages 6–10, 2019.

[23] Lukas Ruff, Jacob R. Kauffmann, Robert A. Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G. Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection, 2020.

[24] Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. In *International Conference on Learning Representations*, 2020.

[25] Gideon Schwarz. Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–464, 03 1978.

[26] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *arXiv preprint arXiv:2007.03898*, 2020.

[27] Robert A. Vandermeulen and Clayton D. Scott. Consistency of robust kernel density estimators. In Shai Shalev-Shwartz and Ingo Steinwart, editors, *COLT 2013 - The 26th Annual Conference on Learning Theory, June 12-14, 2013, Princeton University, NJ, USA*, volume 30 of *JMLR Workshop and Conference Proceedings*, pages 568–591. JMLR.org, 2013.

[28] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2593–2601, 2017.

[29] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Towards deeper understanding of variational autoencoding models. *arXiv preprint arXiv:1702.08658*, 2017.

[30] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2018.

[31] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.

[32] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.