OCDE: Odds Conditional Density Estimator

Alex Akira Okuno [*1] Felipe Maia Polo [*2 3]

Abstract

Conditional density estimation (CDE) models can be useful for many statistical applications, especially because the full conditional density is estimated instead of point estimates, revealing more information about the uncertainty of random variables. In this paper, we propose a new methodology called Odds Conditional Density Estimator (OCDE) to estimate conditional densities in a supervised learning scheme. The main idea is that it is very difficult to estimate $p_{\mathbf{x},\mathbf{y}}$ and $p_{\mathbf{x}}$ in order to have the conditional density $p_{\mathbf{y}|\mathbf{x}}$, but by introducing an instrumental distribution, we transform the CDE problem into a problem of odds estimation, or similarly, training a binary probabilistic classifier. We demonstrate how OCDE works using simulated data and then test its performance against other known state-of-the-art CDE methods in real data. Overall, OCDE is competitive compared with these methods in real datasets.

1. Introduction

Conditional density estimation (CDE) consists of estimating the conditional density of a random vector $\mathbf{y} \in \mathbb{R}^k$ given a random vector $\mathbf{x} \in \mathbb{R}^d$, where generally $k = 1$. It is a difficult problem in the sense that the full conditional distribution $p_{\mathbf{y}|\mathbf{x}}$ has to be estimated for many values of $\mathbf{x}$. In this context, the CDE is a generalization of the regression problem, where instead of estimating only the expectation $\mathbb{E}[\mathbf{y}|\mathbf{x}]$, we model the full conditional distribution, which gives much more information about the uncertainty of the random variable of interest.

In many statistical applications, though, in particular when the conditional distribution $p_{\mathbf{y}|\mathbf{x}}$ is well-behaved, i.e. unimodal and symmetric, $\mathbb{E}[\mathbf{y}|\mathbf{x}]$ is generally informative enough. But, this density is often asymmetric and displays multi-modality, in which case the full conditional density would be much more informative. Besides, having a full density allows us to calculate a diversity of statistical quantities like moments including higher order ones, probability intervals, quantiles, non-trivial expectations and so on.

In this paper, we propose a novel methodology called Odds Conditional Density Estimator (OCDE) in order to address the CDE problem. The idea behind OCDE is that we can transform the problem of conditional density estimation into a classification problem that can be solved by training a binary probabilistic classifier, i.e., neural networks or gradient boosting algorithms.

2. Related Work

Regarding the problem of estimating a conditional density $p_{\mathbf{y}|\mathbf{x}}$ in an i.i.d. context, there have been several previous methodologies in the literature. At first, the CDE was introduced by Rosenblatt (1969), whose approach was to estimate $p_{\mathbf{x},\mathbf{y}}$ and $p_{\mathbf{x}}$ with kernel density estimators. However, kernel methods for density estimations, in general, do not scale well with the dimension of covariates and sample size. Several other papers have developed upon this problem with other approaches such as polynomial regressions (Fan et al., 1996), least squares (Sugiyama et al., 2010) and quantile estimation (Takeuchi et al., 2009).

A recent CDE method that overcomes the aforementioned issues is FlexCoDE (Izbicki & Lee, 2017). This is due to the fact that this method can transform the CDE problem into a regression problem, making the CDE to inherit the properties of the regression method used (for example, variable selection, regularization and good properties in a high-dimensional setting). Our work relates to FlexCoDE in the sense that our method also transforms the CDE problem into a supervised model training problem, but in our case, we need to solve a classification problem.

Another interesting CDE method is NN-KCDE (or nearest-neighbors kernel CDE) (Izbicki et al., 2018). This method is the usual kernel density estimate using only the points closest in covariate space to the target point $\mathbf{x}$ (Izbicki et al., 2018). It is possible to tune the number of neighbors hyper-

[*] Equal contribution [1] Department of Statistics, Institute of Mathematics and Statistics, University of São Paulo, Brazil [2] Department of Statistics, University of Michigan, USA [3] Advanced Institute for Artificial Intelligence (AI2), Brazil. Correspondence to: Alex Akira Okuno <akira.okuno@outlook.com>, Felipe Maia Polo <felipemaiapolo@gmail.com>.

Proceedings of the LatinX in AI (LXAI) Research workshop at ICML 2021. Copyright 2021 by the author(s).

parameter since the method optimizes the CDE Loss.

Our method relies on directly estimating the density ratio $\frac{p_{\mathbf{x},\mathbf{y}}}{p_{\mathbf{x}}}$ using a method called "Probabilistic Classification" or sometimes referred as "Logistic Regression based method" (Sugiyama et al., 2012). Interestingly, Sugiyama et al. (2010) states that "Logistic Regression based method" may not be employed for CDE because $p_{\mathbf{x},\mathbf{y}}$ and $p_{\mathbf{x}}$ do not share the same domains. We get around this problem by introducing an instrumental random variable, which makes the domains match. Given that, our method is an extension of a method for density estimation presented in Section 14.2.4 of Hastie et al. (2009). For the best of our knowledge, a method with this foundation has not been used for conditional density estimation yet.

3. Odds Conditional Density Estimator (OCDE)

In this section, we present the OCDE for multivariate target vector $\mathbf{y}$ and feature vectors $\mathbf{x}$. Suppose we have a set of data points $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ sampled independently from $P_{\mathbf{x},\mathbf{y}}$ with probability density function (p.d.f.) $p_{\mathbf{x},\mathbf{y}}$. Our objective is to estimate $p_{\mathbf{y}|\mathbf{x}}$ using the dataset $\mathcal{D}$. Suppose that $\mathbf{y}$ is a random vector that assumes values in support$(p_{\mathbf{y}}) \subseteq \mathbb{R}^k$. In order to continue, let us introduce an instrumental/artificial random variable $\tilde{\mathbf{y}}$ with known distribution $P_{\tilde{\mathbf{y}}}$, independent from $(\mathbf{x},\mathbf{y})$, with known p.d.f. $p_{\tilde{\mathbf{y}}}$ such that support$(p_{\mathbf{y}}) \subseteq$ support$(p_{\tilde{\mathbf{y}}})$.

See that we can rewrite $p_{\mathbf{y}|\mathbf{x}}$ as follows:

$$p_{\mathbf{y}|\mathbf{x}}(\boldsymbol{y}|\boldsymbol{x}) = \frac{p_{\mathbf{x},\mathbf{y}}(\boldsymbol{x},\boldsymbol{y})}{p_{\mathbf{x}}(\boldsymbol{x})} \tag{1}$$

$$= \frac{p_{\tilde{\mathbf{y}}}(\boldsymbol{y})}{p_{\tilde{\mathbf{y}}}(\boldsymbol{y})} \frac{p_{\mathbf{x},\mathbf{y}}(\boldsymbol{x},\boldsymbol{y})}{p_{\mathbf{x}}(\boldsymbol{x})} \tag{2}$$

$$= p_{\tilde{\mathbf{y}}}(\boldsymbol{y}) \frac{p_{\mathbf{x},\mathbf{y}}(\boldsymbol{x},\boldsymbol{y})}{p_{\mathbf{x},\tilde{\mathbf{y}}}(\boldsymbol{x},\boldsymbol{y})} \tag{3}$$

Our objective of estimating $p_{\mathbf{y}|\mathbf{x}}$ can be reduced in estimating the density ratio $\frac{p_{\mathbf{x},\mathbf{y}}}{p_{\mathbf{x},\tilde{\mathbf{y}}}}$. To that end, we adopt the "Probabilistic Classification" method for density ratio estimation (Sugiyama et al., 2012), which is detailed next. Consider a random vector $(\mathbf{x}', \mathbf{y}') \sim P_{\mathbf{x}',\mathbf{y}'}$ with p.d.f. $p_{\mathbf{x}',\mathbf{y}'}$ and another artificial random variable $\mathrm{s} \sim \mathrm{Bernoulli}(1/2)$, given that

$$p_{\mathrm{x},\mathrm{y}}(\boldsymbol{x},\boldsymbol{y}) = p_{\mathbf{x}',\mathbf{y}'|\mathrm{s}}(\boldsymbol{x},\boldsymbol{y}|\mathrm{s}=1) \tag{4}$$

$$p_{\mathrm{x},\tilde{\mathbf{y}}}(\boldsymbol{x},\boldsymbol{y}) = p_{\mathbf{x}',\mathbf{y}'|\mathrm{s}}(\boldsymbol{x},\boldsymbol{y}|\mathrm{s}=0) \tag{5}$$

That is, s is an indicator variable that tells us if data point comes from $P_{\mathbf{x},\mathbf{y}}$ or $P_{\mathbf{x},\tilde{\mathbf{y}}}$. From Bayes rule, it follows that:

$$p_{\mathbf{y}|\mathbf{x}}(\boldsymbol{y}|\boldsymbol{x}) = p_{\tilde{\mathbf{y}}}(\boldsymbol{y}) \frac{p_{\mathbf{x}',\mathbf{y}'|\mathrm{s}}(\boldsymbol{x},\boldsymbol{y}|\mathrm{s}=1)}{p_{\mathbf{x}',\mathbf{y}'|\mathrm{s}}(\boldsymbol{x},\boldsymbol{y}|\mathrm{s}=0)} \tag{6}$$

$$= p_{\tilde{\mathbf{y}}}(\boldsymbol{y}) \frac{p_{\mathrm{s}|\mathbf{x}',\mathbf{y}'}(\mathrm{s}=1|\boldsymbol{x},\boldsymbol{y})}{p_{\mathrm{s}|\mathbf{x}',\mathbf{y}'}(\mathrm{s}=0|\boldsymbol{x},\boldsymbol{y})} \tag{7}$$

From our original dataset $\mathcal{D}$, we derive another dataset $\tilde{\mathcal{D}} = \{(\boldsymbol{x}_i, \tilde{\boldsymbol{y}}_i)\}_{i=1}^n$, with values $\{\tilde{\boldsymbol{y}}_i\}_{i=1}^n$ being instances of $\tilde{\mathbf{y}}$. Then, we create artificial labels for the data points of $\mathcal{D}$ and $\tilde{\mathcal{D}}$, where the first samples receive labels 1 and the second receive labels 0. In other words, we label samples according to the variable s. Then, we train a *non-linear* binary probabilistic classifier $\widehat{p}_{\mathrm{s}|\mathbf{x}',\mathbf{y}'}$ discriminating samples from $\mathcal{D}$ and $\tilde{\mathcal{D}}$.

Finally, our estimator for the conditional density, OCDE, is given by

$$\widehat{p}_{\mathbf{y}|\mathbf{x}}(\boldsymbol{y}|\boldsymbol{x}) = p_{\tilde{\mathbf{y}}}(\boldsymbol{y}) \frac{\widehat{p}_{\mathrm{s}|\mathbf{x}',\mathbf{y}'}(\mathrm{s}=1|\boldsymbol{x},\boldsymbol{y})}{\widehat{p}_{\mathrm{s}|\mathbf{x}',\mathbf{y}'}(\mathrm{s}=0|\boldsymbol{x},\boldsymbol{y})} \tag{8}$$

Some useful information about OCDE are the following:

- Given that $\widehat{p}_{\mathrm{s}|\mathbf{x}',\mathbf{y}'}$ should be a good estimate for the true conditional distribution $p_{\mathrm{s}|\mathbf{x}',\mathbf{y}'}$, we advise practitioners to optimize the binary cross-entropy/log loss when training or choosing the classifier's hyperparameters. Depending on the model adopted, probability calibration might be necessary;
- It can be the case that $\widehat{p}_{\mathbf{y}|\mathbf{x}}(\boldsymbol{y}|\boldsymbol{x})$ does not integrate to 1 for all possible values of $\boldsymbol{x}$. To fix this problem, we can discretize $\widehat{p}_{\mathbf{y}|\mathbf{x}}(\boldsymbol{y}|\boldsymbol{x})$ into a histogram[1] and then normalize the histogram itself. In practice, we always adopt this strategy;
- The choice of $p_{\tilde{\mathbf{y}}}$ is non-trivial. A standard choice is to assume that $\tilde{\mathbf{y}}$ is uniformly distributed in some reasonable bounded subset of $\mathbb{R}^k$ that can be chosen according to the training set samples;
- In theory, the instrumental random vector $\tilde{\mathbf{y}}$ can be dependent on $\mathbf{x}$, but we do not explore this scenario in this paper.

4. Experiments

4.1. Toy Experiment

In this experiment, we sample from $P_{\mathrm{x},\mathrm{y}}$ indirectly. If $\theta_i \sim U[0, 2\pi]$ and $\epsilon_i \sim N(0,1)$, we use the following functional forms to sample x_i and y_i, for $i = 1, ..., n$:

[1] Histograms of 100 or 1000 bins, for example.

$\mathrm{x}_i = 5\cos(\theta_i)$ and $\mathrm{y}_i = 5\sin(\theta_i) + \epsilon_i$. Given that we sample (θ_i, ϵ_i) independently from (θ_j, ϵ_j), if $i \neq j$, then $(\mathrm{x}_i, \mathrm{y}_i)$ is also independent from $(\mathrm{x}_j, \mathrm{y}_j)$. In this example, we assume the instrumental random variables $\{\tilde{\mathrm{y}}_i\}_{i=1}^{n}$ are sample independently from $U[-10, 10]$, hence having a known p.d.f. In the following, we create the datasets $\mathcal{D}$ and $\tilde{\mathcal{D}}$, with $n = 10000$, and plot them in Figure 1.

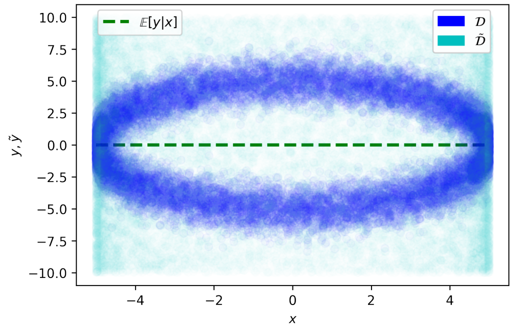

Figure 1. Real ($\mathcal{D}$) and artificial ($\tilde{\mathcal{D}}$) datasets used to estimate $p_{\mathrm{y}|\mathrm{x}}$. We also plot $\mathbb{E}[\mathrm{y}|\mathrm{x}]$ to show it maps to points in the domain of y that, in most cases, have no probability density.

We can make two important observations about Figure 1: (i) for certain values of $\mathrm{x} \in (-5, 5)$, the conditional density $p_{\mathrm{y}|\mathrm{x}}$ is multi-modal and (ii) the expectation $\mathbb{E}[\mathrm{y}|\mathrm{x}]$ maps to points in the domain of y that, in most cases, actually have no probability density.

In this experiment, we use the CatBoost Classifier(Prokhorenkova et al., 2017) with default hyperparameters as our binary probabilistic classifier. Figure 2 lets us see some examples of OCDE estimates for the conditional distribution while varying the values of x. The distinction in the modality of the conditional distribution is very clear when we compare $\widehat{p}_{\mathrm{y}|\mathrm{x}}(y|\mathrm{x} = 0)$ against $\widehat{p}_{\mathrm{y}|\mathrm{x}}(y|\mathrm{x} = 5)$. We do not intend to evaluate our model with this toy experiment, but we have a nice visual intuition that our model incorporates the change of modality reasonably well in a simple setup.

Given that we use the CatBoost Classifier as our binary probabilistic classifier, it is expected that the conditional density $\widehat{p}_{\mathrm{y}|\mathrm{x}}$ is a non-smooth function, which is clear in Figure 2. It is possible to approximate $\widehat{p}_{\mathrm{y}|\mathrm{x}}$ using the Fast Fourier Transform (FFT) algorithm for a smoother estimate, for example.

4.2. Real Datasets Experiments

For the following experiments, 13 regression datasets with no missing values have been selected from two different sources[2]. In these datasets, the target variable is univariate

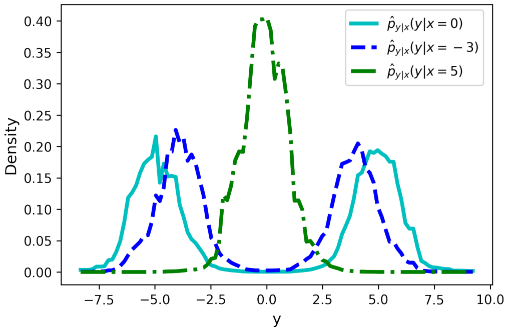

Figure 2. OCDE estimates for the conditional distribution of y while varying the values of x.

while the number of features ranges from 6 to 40. For each one of the 13 datasets, we repeated the following pre-processing steps: (i) we kept up to 5,000 data points per dataset, (ii) normalized each column in every dataset, stretching values in the interval $[0, 1]$, and (iii) randomly splitted the data points in a training set (80%) and a test set (20%). When necessary, we use part of the training set (20%) as a validation set.

For this series of experiments, we take two approaches on estimating conditional density. In the first one (*Raw*), we directly estimate the conditional density of y. In the second one (*Debiased*), we first fit a regressor $\hat{f}(\boldsymbol{x}) = \widehat{\mathbb{E}}[\mathrm{y}|\mathbf{x} = \boldsymbol{x}]$ using the training set and then estimate the conditional density of the residuals $\hat{\varepsilon} = \mathrm{y} - \hat{f}(\mathbf{x})$ given $\mathbf{x}$. In these experiments, the regressor $\hat{f}$ is a CatBoost Regressor (Prokhorenkova et al., 2017).

In each of the two approaches, we compare four alternatives for conditional density estimation: (i) OCDE, (ii) OCDE (Smooth), (iii) FlexCoDE, and (iv) NN-K. Each algorithm has the objective to minimize the CDE Loss (Izbicki & Lee, 2017) on unseen data. More details on each one:

- OCDE: We choose the CatBoost Classifier trained with default hyperparameters and early stopping rounds equals to 50 as our binary probabilistic classifier. The instrumental random variable $\tilde{\mathrm{y}}$ is uniformly distributed in the interval bounded by the minimum and maximum values of the target variable in the training set;
- OCDE (Smooth): This estimator is almost identical to the OCDE. It only differs from the fact that we approximate $\widehat{p}_{\mathrm{y}|\mathrm{x}}$ using the Fast Fourier Transform (FFT) algorithm. The number of Fourier components minimizes the CDE Loss on the training set;
- FlexCoDE: We choose the XGBoost method as the regression method to train FlexCoDE, which was trained with default hyperparameters. Regarding the FlexCoDE itself, we chose 50 as the max basis parameter

[2] https://bit.ly/3h1OeTa, https://bit.ly/3vThlhb

Table 1. CDE Loss (± std. error) obtained by the four alternatives for conditional density estimation in the *Raw* and *Debiased* approaches. In summary, the results presented in this table puts OCDE among the state-of-the-art methods for conditional density estimation.

	Raw				Debiased			
Dataset	OCDE	OCDE (Smooth)	FlexCoDE	NN-K	OCDE	OCDE (Smooth)	FlexCoDE	NN-K
abalone	67.93 ± 1.15	-1.00 ± 0.00	-4.07 ± 0.12	$\mathbf{-8.81 \pm 0.25}$	-4.81 ± 0.03	-4.81 ± 0.03	-4.20 ± 0.06	-4.92 ± 0.05
ailerons	39.14 ± 0.80	-1.12 ± 0.00	-6.40 ± 0.15	-7.81 ± 0.21	$\mathbf{-8.07 \pm 0.03}$	$\mathbf{-8.08 \pm 0.03}$	-7.44 ± 0.06	$\mathbf{-8.08 \pm 0.06}$
bank32nh	$\mathbf{-25.14 \pm 1.19}$	-21.89 ± 1.03	$\mathbf{-26.72 \pm 1.39}$	-11.69 ± 0.42	-5.21 ± 0.22	-5.22 ± 0.23	-4.96 ± 0.19	-4.29 ± 0.28
bank8FM	$\mathbf{-10.54 \pm 0.61}$	-9.73 ± 0.51	-9.44 ± 0.45	-3.08 ± 0.19	$\mathbf{-10.63 \pm 0.30}$	$\mathbf{-10.63 \pm 0.42}$	-8.97 ± 0.32	-9.03 ± 0.32
cal housing	$\mathbf{-6.03 \pm 0.36}$	$\mathbf{-5.77 \pm 0.37}$	$\mathbf{-5.56 \pm 0.18}$	-4.61 ± 0.18	-5.20 ± 0.04	-5.21 ± 0.04	-5.34 ± 0.08	-5.25 ± 0.06
cpu act	$\mathbf{-15.78 \pm 0.79}$	-14.05 ± 0.58	-14.53 ± 0.39	-14.37 ± 0.56	-13.13 ± 0.47	-13.14 ± 0.42	-13.38 ± 0.30	$\mathbf{-14.85 \pm 0.31}$
cpu small	$\mathbf{-14.98 \pm 0.75}$	-13.13 ± 0.65	-13.18 ± 0.34	-13.37 ± 0.41	-12.50 ± 0.31	-12.52 ± 0.32	-12.13 ± 0.28	-13.28 ± 0.29
delta ailerons	-5.48 ± 0.24	-5.69 ± 0.20	-9.21 ± 0.23	$\mathbf{-13.03 \pm 0.37}$	-10.69 ± 0.06	-10.70 ± 0.06	-11.17 ± 0.13	-12.01 ± 0.12
elevators	0.82 ± 0.83	-6.79 ± 0.24	-8.30 ± 0.19	-8.45 ± 0.32	-8.89 ± 0.06	-8.90 ± 0.05	-8.30 ± 0.12	$\mathbf{-9.59 \pm 0.06}$
fried delve	-4.00 ± 0.06	-4.00 ± 0.08	-4.19 ± 0.05	-3.30 ± 0.06	$\mathbf{-8.75 \pm 0.01}$	$\mathbf{-8.75 \pm 0.01}$	-7.82 ± 0.09	-8.66 ± 0.02
puma32H	-3.62 ± 0.04	-3.62 ± 0.05	-4.96 ± 0.07	-1.79 ± 0.05	-5.91 ± 0.03	-5.91 ± 0.03	$\mathbf{-6.16 \pm 0.07}$	-5.80 ± 0.04
puma8NH	-2.28 ± 0.04	-2.28 ± 0.05	-1.89 ± 0.06	-2.06 ± 0.06	$\mathbf{-2.32 \pm 0.03}$	$\mathbf{-2.32 \pm 0.04}$	-2.12 ± 0.04	$\mathbf{-2.37 \pm 0.03}$
winequality	-2.79 ± 1.78	-13.18 ± 1.01	$\mathbf{-26.78 \pm 0.71}$	-19.25 ± 0.47	-2.50 ± 0.03	-2.50 ± 0.03	-14.30 ± 0.24	-2.92 ± 0.07

and used the tuning procedure provided by its Python implementation;

- NN-K: In this estimator, we performed an extensive hyperparameter optimization, specifically for the number of neighbors and bandwidth level.

Table 1 compares the CDE Loss (± std. error), estimated on the test set, obtained by the four alternatives for conditional density estimation in the *Raw* and *Debiased* approaches. In summary, the results presented in Table 1 puts OCDE among the state-of-the-art methods for conditional density estimation. Considering that two methods have the same performance if their error bars intercept, it is possible to see that OCDE has the best results in 8 datasets, while NN-K has the best results in 6 datasets, OCDE (Smooth) is the best in 5 datasets, and FlexCoDE is the best in 4 of them.

The average running times ± std. deviation (in seconds) for the four methods are the following: (i) OCDE: 2.26 ± 1.97; (ii) OCDE (Smooth): 128.96 ± 47.59; (iii) FlexCoDE: 125.18 ± 76.5; (iv) NN-K: 27.36 ± 2.66.

5. Conclusion and Future Work

In this paper, we propose a novel methodology called Odds Conditional Density Estimator (OCDE) in order to address the CDE problem. OCDE performed well against competitors in real dataset experiments. Future directions for this work could be: (i) testing how OCDE performs with different sample sizes and target dimensions, (ii) optimizing the choice of the instrumental random vector $\hat{\mathbf{y}}$, or (iii) testing OCDE against other benchmark approaches.

References

Fan, J., Yao, Q., and Tong, H. Estimation of conditional densities and sensitivity measures in nonlinear dynamical systems. *Biometrika*, 83(1):189–206, 1996.

Hastie, T., Tibshirani, R., and Friedman, J. *The elements of statistical learning: data mining, inference, and prediction.* Springer Science & Business Media, 2009.

Izbicki, R. and Lee, A. Converting high-dimensional regression to high-dimensional conditional density estimation. *Eletronic Journal of Statistics*, 2017.

Izbicki, R., Lee, A. B., and Pospisil, T. Abc-cde: Towards approximate bayesian computation with complex high-dimensional data and limited simulations. *arXiv preprint arXiv:1805.05480*, 2018.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. Catboost: unbiased boosting with categorical features. *arXiv preprint arXiv:1706.09516*, 2017.

Rosenblatt, M. Conditional probability density and regression estimators. In Krishnaiah, P. (ed.), *Multivariate Analysis II*. 1969.

Sugiyama, M., Takeuchi, I., Suzuki, T., Kanamori, T., Hachiya, H., and Okanohara, D. Least-squares conditional density estimation. *IEICE Transactions on Information and Systems*, 93(3):583–594, 2010.

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

Takeuchi, I., Nomura, K., and Kanamori, T. Nonparametric conditional density estimation using piecewise-linear solution path of kernel quantile regression. *Neural Computation*, 21(2):533–559, 2009.