Calibration: A Simple Way to Improve Click Models

Alexey Borisov
Yandex & University of Amsterdam
alborisov@yandex-team.ru

Julia Kiseleva*
UserSat.com & University of Amsterdam
j.kiseleva@uva.nl

Ilya Markov
University of Amsterdam
i.markov@uva.nl

Maarten de Rijke
University of Amsterdam
derijke@uva.nl

ABSTRACT

We show that click models trained with suboptimal hyperparameters suffer from the issue of bad calibration. This means that their predicted click probabilities do not agree with the observed proportions of clicks in the held-out data. To repair this discrepancy, we adapt a non-parametric calibration method called *isotonic regression*. Our experimental results showthat isotonic regression significantly improves click models trained with suboptimal hyperparameters in terms of perplexity, and that it makes click models less sensitive to the choice of hyperparameters. Interestingly, the relative ranking of existing click models in terms of their predictive performance changes depending on whether or not their predictions are calibrated. Therefore, we advocate that calibration becomes a mandatory part of the click model evaluation protocol.

CCS CONCEPTS

• **Information systems → Users and interactive retrieval**;

KEYWORDS

Click models, Calibration

ACM Reference format:
Alexey Borisov, Julia Kiseleva, Ilya Markov, and Maarten de Rijke. 2018. Calibration: A Simple Way to Improve Click Models. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Turin, Italy, October 22–26, 2018 (CIKM 2018)*, 4 pages.
https://doi.org/10.1145/3269206.3269260

1 INTRODUCTION

Click models [3] are important and widely used tools for interpreting user behavior in Web search. A common way to evaluate their performance is to measure how well they predict clicks on the documents presented on a *search engine result page* (SERP). As for many machine learning algorithms, the performance of click models strongly depends on the hyperparameters used for training.[1]

We hypothesize that click models trained with suboptimal hyperparameters are often not well *calibrated*. This means that their predicted click probabilities do not agree with the observed proportions of clicks in the held-out data. We validate how well a click model is calibrated using a reliability diagram [14]. For each rank, we split query sessions into $N = 100$ buckets according to the predicted click probabilities, where the i-th bucket corresponds to click probabilities in the range from $\frac{i}{N}$ to $\frac{i+1}{N}$, and plot the observed *click-through rate*s (CTRs) in these buckets. For a well-calibrated click model, the observed CTRs in each bucket should lie in the range of the predicted click probabilities associated with this bucket. Fig. 1 shows a reliability diagram of the *click chain model* (CCM)

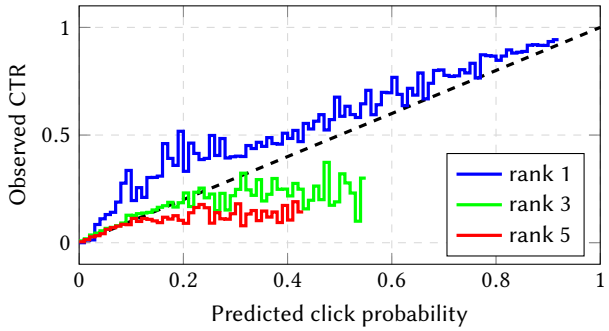

Figure 1: Reliability diagram of CCM at ranks 1, 3 and 5.

trained on a publicly available dataset (see §4 for details) with suboptimal hyperparameters.[2] We learn from Fig. 1 that CCM tends to underestimate click probabilities at rank 1 and overestimate click probabilities at ranks 3 and 5. We observe similar trends for other click models and therefore conclude that click models suffer from the issue of bad calibration.

There are two approaches to calibration: parametric and non-parametric [15]. The parametric approach is less flexible but also requires less data for calibration. The non-parametric approach is more general but requires more calibration data. Since real-world click logs are large, we follow the latter approach. In particular, we propose to use *isotonic regression* [17] to repair the discrepancy between the click probabilities predicted by a model and the proportion of clicks in the held-out data.

Our experiments show that(i) isotonic regression significantly improves click models trained with suboptimal hyperparameters in terms of perplexity; and that (ii) calibrated click models are less sensitive to the choice of hyperparameters than the original (non-calibrated) ones.

2 BACKGROUND AND RELATED WORK

Assessment of probabilities. Estimating probabilities of future events is important for effective decision making [13, 18]. Studies

*Now at Microsoft Research AI.
[1]This statement is based on our preliminary experiments with a range of click models and their hyperparameters. See §4 for details.

[2]For rank 1, the prior values of the CCM parameters are distributed according to Beta(1, 10); for ranks 3 and 5, according to Beta(1, 2).

CIKM 2018, October 22–26, 2018, Turin, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3269260

show that people are generally not good at these tasks [13]. They tend to overestimate or underestimate their confidence, which introduces biases in their predictions [13].

Recent work demonstrates that many popular machine learning methods also suffer from the issue of bad calibration [5, 15, 16, 20, 21]. Niculescu-Mizil and Caruana [15] show that under unrealistic independence assumptions, Naive Bayes tends to produce probabilities that are too close to the extreme values of 0 and 1 (see [5] for a theoretical analysis), while SVM and boosted decision trees rarely predict probabilities that are close to 0 and 1. This suggests that predictions of these learning algorithms are biased.

To alleviate the discrepancy between probabilities predicted by a binary classifier and observed frequencies, both parametric and non-parametric calibration methods have been investigated. Platt [16] puts forward the idea of fitting a sigmoid transformation between the outputs predicted by the binary classifier and the observed labels. Zadrozny and Elkan [21] suggest using isotonic regression [17], which learns a monotone transformation of the scores computed by the binary classifier to probabilities of class I. Niculescu-Mizil and Caruana [15] recommend using Platt scaling when the data used for calibration is limited and isotonic regression when there is enough data for calibration.

Click modeling. Click data is a valuable signal for improving Web search [2, 10]. However, accurately interpreting user clicks on a SERP is not straightforward due to the so-called *position bias effect* [8, 11]: people tend to click more on the documents presented on top positions than on the documents presented on lower positions. To account for this and other types of bias, click models have been proposed [3].

Traditional click models consist of Bernoulli-distributed random variables $X \sim \text{Bernoulli}(\theta)$ associated with query-document pairs [3]. Here, θ denotes a parameter associated with the query-document pair, e.g., *attractiveness* (i.e., the probability of a user examining the document's snippet) and *satisfactoriness*, (i.e., the probability of a user's information need being satisfied after interacting with the document). The value of the parameter θ is estimated during training. It is initially specified by a Beta distribution, $\theta_{\text{prior}} \sim \text{Beta}(\alpha, \beta)$, and then updated upon observing new click/skip data. The choice of the training hyperparameters α and β impacts the overall performance of click models, especially in query sessions that contain rare or previously unseen query-document pairs. Existing work on click models rarely provides sufficient details on tuning click model hyperparameters, even when they aim to systematically compare click models [7], which is troublesome because experiments without properly tuned hyperparameters may yield misleading results [3].

The key distinction of our work compared to the work listed above is that we are the first to improve the performance of click models by applying calibration.

3 METHOD

Click models are trained to predict probabilities of a user clicking on the ranked list of documents $d_1, \ldots, d_n$ returned by a search engine in response to a user's query q. Click models utilize different assumptions about how a user interacts with $d_1, \ldots, d_n$. E.g., many click models make the *linear traversal assumption* [4], which states that a user examines documents on a SERP from top to bottom. Such models predict the probability of observing a click on document d_{r+1} given a user's query q and clicks $c_1, \ldots, c_r$ on the higher ranked documents:

$$P(c_{r+1} = 1 \mid q, d_1, \ldots, d_{r+1}, c_1, \ldots c_r), \tag{1}$$

where $c_i = 1$ if a user clicked on document d_i, and 0 otherwise. In order to compare click models, which follow different assumptions about user click behavior, conditional click probabilities, presented in Eq. (1), are marginalized to click probabilities $P(c_{r+1} = 1 \mid q, d_1, \ldots, d_{r+1})$ that are unconditional on previous clicks:

$$\sum_{(c_1, \ldots, c_r)} P(c_{r+1} = 1 \mid q, d_1, \ldots, d_{r+1}, c_1, \ldots, c_r), \tag{2}$$

where the sum is computed over all possible click combinations on the first r documents.

As discussed in §1, for existing click models the click probabilities shown in Eq. (1) and Eq. (2) do not represent the observed CTRs well and, thus, need to be calibrated. We calibrate these probabilities separately for each rank. Following the recommendations in [15] and considering that real-world click data is usually available in large quantities, we adopt a non-parametric calibration method, namely isotonic regression [17]. For each rank r, isotonic regression learns a function $g_r(P)$ that adjusts the click probabilities predicted at rank r. Specifically, it solves the following optimization problem:

$$g_r^* = \arg\min_{g \in \mathcal{G}} \sum_{i=1}^{N} [g(P_r(s^i)) - c_r^i]^2, \tag{3}$$

where $\mathcal{G}$ denotes the set of all piecewise linear, isotonic (non-decreasing), continuous functions, N denotes the number of query sessions used for calibration, $P_r(s^i)$ denotes the predicted click probability at rank r in query session s^i and c_r^i denotes whether the user clicked on the document at rank r in query session s^i.

We use the *pair-adjacent violators* (PAV) algorithm [1] to find the optimal function $g_r^*(P)$. This is done in three steps, illustrated in Fig. 2. First, we sort query sessions s^i by the predicted click probabilities at rank r:

$$P_r(s^{i-1}) \leq P_r(s^i) \qquad \forall i = 2, \ldots, N. \tag{4}$$

We use red dots to display the output of this step in Fig. 2.

Second, we fit a piecewise linear function $g_r(P)$ to the sorted sequence of pairs $[P_r(s^1), c_r^1], \ldots, [P_r(s^N), c_r^N]$ in the following way. For P that is lower than $P_r(s^1)$, i.e., lower than the first click probability in the sorted sequence, $g_r(P)$ returns zero. For P that is larger than $P_r(s^N)$, i.e., larger than the last click probability in the sorted sequence, $g_r(P)$ returns the value of the last click c_r^N. For P that is between two consecutive $P_r(s^{i-1})$ and $P_r(s^i)$, $g_r(P)$ returns the value of click c_r^{i-1}. Formally, this can be written as follows:

$$g_r(P) = \begin{cases} 0 & P < P_r(s^1) \\ c_r^{i-1} & P \in [P_r(s^{i-1}), P_r(s^i)) \quad \forall i = 2, \ldots, N \\ c_r^N & P \geq P_r(s^N). \end{cases} \tag{5}$$

The piecewise linear function $g_r(P)$ calculated at this step is shown with the blue line in Fig. 2.

Third, if the above $g_r(P)$ is not isotonic, there exist two consecutive query sessions s^{i-1} and s^i for which $g_r(P)$ decreases (instead of increasing or staying constant), i.e., $g_r(P_r(s^{i-1})) > g_r(P_r(s^i))$. Such query sessions are called *pair-adjacent violators*. In this case, we change the value of $g_r(P)$ for the interval $P \in [P_r(s^{i-1}), P_r(s^{i+1}))$ to the average of $g_r(P_r(s^{i-1}))$ and $g_r(P_r(s^i))$.[3] This way, for pair-adjacent violators $g_r(P)$ does not decrease anymore, but stays constant and equal to the above-mentioned average. This averaging process is performed in the direction from $P_r(s^1)$ to $P_r(s^N)$. In the

[3] If $i = N$, we perform this averaging for $P \geq P_r(s^{N-1})$.

end, $g_r(P)$ becomes isotonic as shown with the green line in Fig. 2. See [1] for a proof of the optimality of $g_r(P)$ with respect to Eq. (3).

Note that the optimal calibration function $g_r^*(P)$ learned by PAV outputs a probability of 0 for $P < P_r(s^1)$ and might output a probability of 1 for large values of P. Following [15, 16] and to avoid problems with taking the log of $g_r^*(P)$, we trim $g_r^*(P)$ to predict probabilities in the range $[\delta, 1-\delta]$ instead of the range $[0, 1]$. In our experiments, we use $\delta = 0.01$.

Now that we have learned a calibration function $g_r^*(P)$ for a click model, the calculation of click probabilities at rank r works as follows. Given a query session s, the click model predicts the click probability $P_r(s)$ for rank r in that session. This could be either the conditional, Eq. (1), or unconditional, Eq. (2), probability.[4] The predicted probability $P_r(s)$ is then passed to the calibration function $g_r^*(P)$, which outputs the calibrated probability. This calibrated probability is then used for click prediction.

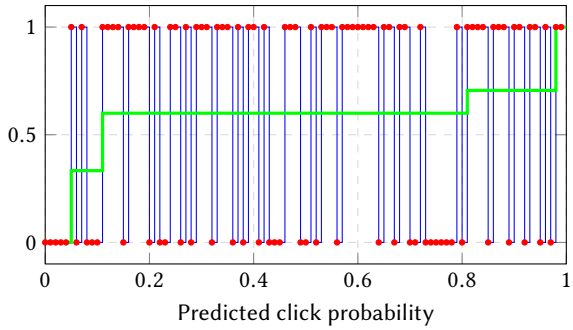

Figure 2: Illustration of the three steps of the PAV algorithm. Red dots represent observed clicks vs. predicted click probabilities. The blue line represents the piece-wise linear function in Eq. (5). The green line represents the learned isotonic transformation from original click probabilities to calibrated click probabilities.

4 EXPERIMENTAL SETUP

We design our experiments to answer two research questions:

RQ1 *Does isotonic regression help to improve the performance of existing click models?*

RQ2 *Does isotonic regression make click models less dependent on the choice of hyperparameters?*

Dataset and evaluation methodology. We conduct our experiments using a publicly available dataset released for the Yandex Relevance Prediction challenge by Yandex, the major search engine in Russia.[5] Query sessions are ordered by time. We use the first 1,000,000 query sessions as the *training set*, the following 100,000 query sessions as the *development set* and the next 100,000 query sessions as the *test set*.

We evaluate click models using perplexity [3], which measures how "surprised" a model is upon observing a particular set of clicks on a SERP. We calculate perplexity at position k as follows:

$$\text{Perplexity@k} = 2^{-\frac{1}{N}\sum_{i=1}^{N}\log_2 P(c_k=c_k^i|q^i,d_1^i,\ldots,d_n^i)}, \quad (6)$$

where N denotes the number of query sessions in the test set; q^i is a query in the i-th session; $d_1^i, \ldots, d_n^i$ are documents retrieved by a search engine in the i-th session in response to the query q_i; $c_k^i = 1$ if a user clicked on the document and 0 otherwise. Following [6], we use perplexity averaged over all positions as our main metric. Lower values of perplexity correspond to higher quality of a model. For each click model, we perform significance testing using a paired t-test on the perplexity scores computed using different sets of hyperparameters. Differences are considered statistically significant for p-values lower than 0.05. We do not evaluate click models on the relevance prediction task [3], because the inferred query-document-specific parameters used for ranking are not affected by the proposed calibration method.

Experiment 1. To answer RQ1, we measure, before and after calibration, the average of the perplexity values computed for a click model $\mathcal{M}$ trained with different hyperparameters. If the average value of perplexity is lower after calibration, we conclude that calibration helps to improve the performance of $\mathcal{M}$. Otherwise, we conclude that calibration does not help or even hurts the performance of $\mathcal{M}$.

Experiment 2. To answer RQ2, we measure, before and after calibration, the variance of the perplexity values computed for a click model $\mathcal{M}$ trained with different hyperparameters. If the variance of the perplexity values is lower after calibration, we conclude that calibration makes $\mathcal{M}$ less dependent on the choice of hyperparameters. Otherwise, we conclude that calibration does not make M less dependent on the choice of hyperparameters or even makes it more sensitive to the choice of hyperparameters.

We conduct our experiments using four *probabilistic graphical models* (PGMs) that are often used for modeling and predicting clicks on a SERP: the *dynamic Bayesian network* (DBN) [2], the *dependent click model* (DCM) [9], the *click chain model* (CCM) [9], and the *user browsing model* (UBM) [6]. We train these click models by maximizing the likelihood of the observed click/skip events in our logs. For DCM we optimize the likelihood directly, and for DBN, CCM and UBM we use the *expectation-maximization* (EM) algorithm with 50 iterations. We set the prior values of the parameters of these click models to $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{4}$, $\frac{1}{5}$, $\frac{1}{6}$, $\frac{1}{7}$, $\frac{1}{8}$, $\frac{1}{9}$, $\frac{1}{10}$, $\frac{2}{3}$, $\frac{2}{4}$, $\frac{2}{5}$, $\frac{2}{6}$, $\frac{2}{7}$, $\frac{2}{8}$, $\frac{2}{9}$, $\frac{2}{10}$, $\frac{3}{4}$, $\frac{3}{5}$, $\frac{3}{6}$, $\frac{3}{7}$, $\frac{3}{8}$, $\frac{3}{9}$, $\frac{3}{10}$.

5 RESULTS

In this section, we present the results of our experiments and provide answers to our research questions.

Experiment 1. Table 1 shows the perplexity of the click models CCM, DBN, DCM and UBM averaged over ranks and over runs with different hyperparameters. The rows of Table 1 correspond to: (i) *Baseline w/o dev set*, a method where click models are trained on the training set; (ii) *Baseline w/ dev set*, a method where click models are trained on the union of the training set and the development set; (iii) *Calibrated*, a method where click models are trained on the training set and calibrated on the development set.

From Table 1, we conclude that isotonic regression improves the performance of the selected click models. The differences in performance between the click models trained(i) on the training set only, and (ii) on the union of the training set and the development set (Table 1, row 2 vs. row 1) are much less than the gains achieved from using the development set for calibration (Table 1, row 3 vs. row 1). This means that the gains obtained from the calibration method described in §3 are not (only) due to using more data (i.e., the development set), but are due to fixing the miscalibration problem.

[4] Note that calibration should be done separately for each of those probabilities.
[5] http://imat-relpred.yandex.ru/en/datasets (last visited August 25, 2018).

Table 1: Perplexity of click models with and without calibration averaged over ranks and over runs with different hyperparameters. Improvements of the proposed calibration method over both baselines are statistically significant ($p < 0.001$). The best results are given in bold face.

		Average perplexity			
#	Method	CCM	DBN	DCM	UBM
1	Baseline w/o dev set	1.3896	1.3915	1.3826	1.3724
2	Baseline w/ dev set	1.3890	1.3908	1.3822	1.3719
3	**Calibrated**	**1.3722**	**1.3705**	**1.3752**	**1.3659**

Interestingly, the relative ranking of click models in terms of perplexity differs, depending on whether we use calibration or not. From Table 1, we infer the following rankings:

$$\text{UBM} > \text{DCM} > \text{CCM} > \text{DBN} \quad \text{(w/o calibration)} \quad (7)$$

$$\text{UBM} > \text{DBN} > \text{CCM} > \text{DCM} \quad \text{(w/ calibration)} \quad (8)$$

where we write $\mathcal{M}_X > \mathcal{M}_Y$ to denote that click model $\mathcal{M}_X$ has better prediction performance (in terms of perplexity) than click model $\mathcal{M}_Y$. Intuitively, the results w/ calibration make more sense, because DBN and CCM make more realistic assumptions than DCM, which assumes (i) that a user's information need cannot be satisfied directly on a SERP (i.e., a user needs to clicks at least one document presented on the SERP); and (ii) that the probability of examining the document at rank $(r + 1)$ after clicking on the document presented at rank r depends solely on the rank r and not on the relevance of the document presented at rank r.

Answering RQ1, we conclude that the calibration method described in §3 provides good means to fix the miscalibration problem and, as a result, allows us to improve the performance of existing click models on the standard click prediction task.

Experiment 2. Tables 2 and 3 show the empirical variance in the perplexity values computed for click models trained with different sets of hyperparameters. Table 2 lists the absolute values; Table 3 lists the percentages w.r.t. the baseline w/o dev set. For the methods, we use the same naming conventions as in Table 1. We find that the calibration method described in §3 reduces the variance in the perplexity values by 33.87%–98.25%, depending on the click model.

Answering RQ2, we conclude that the calibration method described in §3 makes click models less dependent on the choice of hyperparameters.

6 CONCLUSIONS AND FUTURE WORK

We introduced the notion of calibration in the context of click modeling. We showed empirically that existing click models are prone to produce poorly calibrated predictions and that calibration, namely isotonic regression, (i) improves the performance of click models, and (ii) makes click models less sensitive to tuning of hyperparameters. Therefore, we advocate that calibration becomes a mandatory part of the click model evaluation protocol. In future work, we are planning to incorporate calibration at training time, e.g., by means of hierarchical priors [19] or variational auto-encoders [12].

Table 2: The empirical variance in the perplexity values computed for click models trained with different sets of hyperparameters. The best results are given in bold face.

		$100 \times$ variance in perplexity			
#	Method	CCM	DBN	DCM	UBM
1	Baseline w/o dev set	0.1025	0.1113	0.0908	0.0629
2	Baseline w/ dev set	0.1001	0.1087	0.0886	0.0606
3	**Calibrated**	**0.0079**	**0.0179**	**0.0619**	**0.0011**

Table 3: The empirical variance in the perplexity values computed for click models trained with different sets of hyperparameters. The best results are given in bold face.

		Variance in perplexity			
#	Method	CCM	DBN	DCM	UBM
1	Baseline w/o dev set	100%	100%	100%	100%
2	Baseline w/ dev set	97.66%	97.66%	97.58%	96.34%
3	**Calibrated**	**7.71%**	**16.08%**	**68.17%**	**1.75%**

Acknowledgments. This research was partially supported by Amsterdam Data Science, the Google Faculty Research Awards program, the Microsoft Research Ph.D. program, the Netherlands Institute for Sound and Vision, and the Netherlands Organisation for Scientific Research (NWO) under project nr CI-14-25. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] M. Ayer, H.D. Brunk, G.M. Ewing, W.T. Reid, and E. Silverman. 1955. An empirical distribution function for sampling with incomplete information. *The Annals of Mathematical Statistics* (1955), 641–647.
[2] O. Chapelle and Y. Zhang. 2009. A dynamic bayesian network click model for web search ranking. In *WWW*. ACM, 1–10.
[3] A. Chuklin, I. Markov, and M. de Rijke. 2015. *Click Models for Web Search*. Morgan & Claypool.
[4] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. 2008. An experimental comparison of click position-bias models. In *WSDM*. ACM, 87–94.
[5] P. Domingos and M. Pazzani. 1996. Beyond independence: Conditions for the optimality of the simple Bayesian classifier. In *ICML*. Morgan Kaufm., 105–112.
[6] G.E. Dupret and B. Piwowarski. 2008. A user browsing model to predict search engine click data from past observations. In *SIGIR*. ACM, 331–338.
[7] A. Grotov, A. Chuklin, I. Markov, L. Stout, F. Xumara, and M. de Rijke. 2015. A comparative study of click models for web search. In *CLEF*. Springer, 78–90.
[8] Z. Guan and E. Cutrell. 2007. An eye tracking study of the effect of target rank on web search. In *CHI*. ACM, 417–420.
[9] F. Guo, C. Liu, and Y.M. Wang. 2009. Efficient multiple-click models in web search. In *WSDM*. ACM, 124–131.
[10] T. Joachims. 2002. Optimizing search engines using clickthrough data. In *KDD*. ACM, 133–142.
[11] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. 2005. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR*. ACM, 154–161.
[12] D.P. Kingma and M. Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
[13] S. Lichtenstein, B. Fischhoff, and L.D. Phillips. 1977. Calibration of probabilities: The state of the art. In *Decision Making and Change in Human Affairs*. Springer, 275–324.
[14] A.H. Murphy and R.L. Winkler. 1977. Reliability of subjective probability forecasts of precipitation and temperature. *Applied Statistics* 26, 1 (1977), 41–47.
[15] A. Niculescu-Mizil and R. Caruana. 2005. Predicting good probabilities with supervised learning. In *ICML*. ACM, 625–632.
[16] J.C. Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*. MIT Press, 61–74.
[17] T. Robertson, F.T. Wright, and R. Dykstra. 1988. *Order Restricted Statistical Inference*. Wiley.
[18] P. Slovic. 2016. *The Perception of Risk*. Routledge.
[19] M. West and M.D. Escobar. 1993. *Hierarchical Priors and Mixture Models, with Application in Regression and Density Estimation*. Technical Report. CMU.
[20] B. Zadrozny and C. Elkan. 2001. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *ICML*. ACM, 609–616.
[21] B. Zadrozny and C. Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *KDD*. ACM, 694–699.