A Features-based CTR Probability Prediction using Automatic Differentiation

Soroosh Shalileh

June 2020

1 Proposed Method

1.1 Feature-based Classification using Automatic Differentiation

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ be the set of N pairs of data points x_i and its corresponding label y_i, such that $x_i \in \mathbb{R}^{V \times M}$, and $y_i \in \{-1, 1\}$. Clearly, when M is equal to unity the ordinary form of an entity-to-feature matrix will be formed. And let θ_v represents the underlying distribution of v-th feature of a V-dimensional data points ($v = 1, 2, ..., V$).

Given the aforementioned data set $\mathcal{D}_v$ a restricted version of $\mathcal{D}$ at the v-th feature, the likelihood function can be defined as follows:

$$p(\mathcal{D}_v|\theta_v) = \prod_{i=1}^{N} p(y_i|x_{iv}, \theta_v) \tag{1}$$

It is common to compute the negative of log of likelihood (Eqn.(1)). Therefore, taking negative of logarithm of this equation implies:

$$-\log p(\mathcal{D}_v|\theta_v) = -\sum_{i=1}^{N} \log p(y_i|x_{iv}, \theta_v) \tag{2}$$

Since $\log p(\mathcal{D}_v|\theta_v)$ is a function of θ_v let us denote it with $\mathcal{L}(\theta_v)$.To utilize the benefits of Automatic Differentiation, we do not limit ourselves to a specific model and we try to optimize this equation with its current form.

Applying first order optimaliy on Eqn.(2) implies:

$$\nabla_{\theta_v} \mathcal{L}(\theta_v) = -\nabla_{\theta_v} [\sum_{i=1}^{N} \log p(y_i|x_{iv}, \theta_v)] \tag{3a}$$

$$= -\sum_{i=1}^{N} [\nabla_{\theta_v} \log p(y_i|x_{iv}, \theta_v)] \tag{3b}$$

Due to dominated convergence theorem, we can push the derivative inside the summation. The rest of the obtained results are obvious. Disregarding the model we can easily use AD techniques to compute $\nabla_{\theta_v} \log p(y_i|x_{iv}, \theta_v)$.

We summarized our proposed method for optimizing Eqn.(3) in algorithm(1). It is note worthy to add that in this work we use Adam optimizer [8] while applying other optimizer is also pissible.

Algorithm 1: Feature-based classifier with BBVI and AD

Input: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$: training set
Hyper-parameters: α: learning rate
Result: θ_v, ϕ_v: learned parameters of v-th feature
while *not converged* **do**
  $\mathcal{M} = \{x_{iv}, y_i\}_{i=1}^{M} \sim \mathcal{D}$; % Draw mini-batch of samples $\mathcal{M}$ from $\mathcal{D}$
  $\nabla_{\theta_v} \leftarrow \sum_{i=1}^{M} \nabla_{\theta_v} \log p(y_i|x_{iv}, \theta_v)$; % where $y_i, x_{iv} \in \mathcal{M}$
  $\theta_v \leftarrow optimizer(\nabla_{\theta_v})$;
end

1.2 Features-based CTR Probability Prediction Algorithm

In order to determine which set of feature(s) are critical to predict the probability of occurring a "click" on an advertisement there might be several approaches to check this. For instance, one could define a Euclidean/Minkowski measure to check these condition(s). While someone else could apply totally different approach. And obviously, each has some advantages and disadvantages. In this work, we decided to utilize neural networks to model the features or the combinations of them; and after optimizing, since we expect to obtain a high prediction probability for either of the classes, if the predicted probability is not as high as it is expected and this situation occurs for all features and for all possible combination of them, *we can draw a conclusion that either the labels are not correct or we are facing limits of the current method and/or optimization approaches.*

Our proposed classification method described in subsection(1.1) has the four following advantages 1) it does not require any further mathematical manipulation regarding the adopted model(model-free); 2) it takes into account the underlying distributions of data points (instead of assuming them i.i.d data points); 3) it benefits the advances of AD; 4) It generally faster than many other methods (experiments/justification are needed).

Our proposed Feature-based CTR Probability Prediction Algorithm contains a user-defined hyperparameters, namely, τ which is a threshold over predictions' probability. At first, we start to train our classifier per each feature on a given training set. In the case that only normal data are available, a method like [4] can be adopted. Once the training procedure is done, we utilize the obtained parameters θ_v, to compute the prediction probability per each data point in the test set. And we choose the one with higher probability, and if the predicted probability is greater than τ, the predicted label of that data point will be

accepted and that data point will be removed from the test set otherwise this procedure will be applied with other features.

Once the above procedure is applied for all of the features, we check the test set, if the test set is not empty, we will repeat this procedure for the all possible combination of two features, and then we will check the emptiness of the test set: if still there are some remaining data points in it, this time, we repeat the procedure for all possible combinations of three features and so on. In two cases, the algorithm should halt 1) the test set is empty, or 2) all the combinations of features have been checked. In the latter case, if, still, there are some unclassified data points in the test set, we assign the ambiguous label to those remaining data points [1].

Algorithm (2) summarizes our proposed feature-based CTR probability prediction detection method. In this algorithm, for the j-th data point and c-th feature (or combination of features) let p_{jc}^{+1} represents the probability of that data point being clicked. And it is computed as follows $p_{jc}^{+1} = p(y_{jc} = 1|x_{jc}, \theta_c)$,

[1] In such a case, we face with a limit in Machine Learning with worths further investigations.

similarly, p_{jc}^{-1} represents the probability of being not clicked.

Algorithm 2: FADA Features-based CTR probability Prediction Algorithm

Input:
$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$; % training set
$E = \{(x_j, y_j)\}_{j=1}^{M}$; % test set
Hyper-parameters: τ: a threshold over predictions' probability
Result: $K = \{k_i\}_{i=1}^{N}$ where $k_i \in \{-1, 0, 1\}$: i-th data point predicted label (-1= abnormal, 1=normal, 0=ambiguous)
$V = \{v_1, v_2, ..., v_V\}$ % set of features
$I = E$
For v in **Range(len(V)):** % V is the number features/variable
{ C } = combinations(V, v); % v-th length combination of set V
For c in **{C}:**
$\Omega_c \leftarrow$ *apply Alg.* (1) *on* $\mathcal{D}$; % classifier parameters
For j in **{E}:**
% compute normal probability:
$p_{jc}^{+1} = \log p(y_{jc} = 1 | x_{jc}, \Omega_c)$;
% compute abnormal probability:
$p_{jc}^{-1} = \log p(y_{jc} = -1 | x_{jc}, \Omega_c)$
$p_j = \max(p_{jc}^{-1}, p_{jc}^{+1})$
If $p_j \geq \tau$ **:**
$y_j := argmax(p_j^{-1}, p_j^{+1})$
$\{E\} = \{E\} \setminus \{j\}$
$A = \{I\} \setminus \{E\}$
If $A == \emptyset$ **:** % after iterating over all the data points in test set
Halt
If $A \neq \emptyset$ **:** % after examining all the possible combinations of features
$\{y_a = 0; \forall a \in A\}$ % data point with ambiguous labels
Halt

References

[1] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: a survey. arXiv preprint arXiv:1502.05767, 2015

[2] C.M. Bishop, 2006. Pattern recognition and machine learning. springer.

[3] D. Blei, R. Ranganath, and S. Mohamed, 2016. Variational Inference: Foundations and Modern Methods. Neural Information Processing Systems (NIPS) Tutorial.

[4] Borisyak, M., Ryzhikov, A., Ustyuzhanin, A., Derkach, D., Ratnikov, F. and Mineeva, O., 2019. $(1+\varepsilon)$-class Classification: an Anomaly Detection Method for Highly Imbalanced or Incomplete Data Sets. arXiv preprint arXiv:1906.06096.

[5] Cinlar, E., 2011. Probability and stochastics (Vol. 261). Springer Science & Business Media.

[6] D. R. Cox and D.V. Hinkley.Theoretical Statistics.Chapman and Hall, 1979

[7] D.P Kingma, and M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv, 2013, 1312.6114.

[8] Kingma, D.P. and Ba, J., A method for stochastic optimization,arXiv preprint arXiv, 2014, cs.LG, 1412.6980

[9] Minka, T., & others. (2005), Divergence measures and message passing. Technical report, Microsoft Research.

[10] Mnih, V., Heess, N. and Graves, A., 2014. Recurrent models of visual attention. In Advances in neural information processing systems (pp. 2204-2212).

[11] Mohamed, S., Rosca, M., Figurnov, M. and Mnih, A., 2019. Monte carlo gradient estimation in machine learning. arXiv preprint arXiv:1906.10652.

[12] Murphy, K.P., 2012. Machine learning: a probabilistic perspective. MIT press.

[13] Ranganath, R., Gerrish, S. and Blei, D.M., 2014. Black Box Variational Inference. In Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics.

[14] Turner, R. E., & Sahani, M. (2011). Two problems with variational expectation maximisation for time-series models. Cambridge University Press.