商品レビューの有用性ランキング推定

柴田, 知親
九州大学大学院システム情報科学府

伊東, 栄典
九州大学情報基盤研究開発センター

出版情報：SIG-KBS. B5 (02), pp.25-30, 2019-11-10. 人工知能学会
バージョン：

商品レビューの有用性ランキング推定

Ranking prediction for product review helpfulness

柴田 知親 [1*]　伊東 栄典 [2†]
Tomochika Shibata[1] and Eisuke Ito[2]

[1] 九州大学大学院システム情報科学府
[1] Graduate School of ISEE, Kyushu University
[2] 九州大学情報基盤研究開発センター
[2] Research Institute for IT, Kyushu University

Abstract: Product reviews can be helpful for consumers to finalize their purchasing decisions. The advertisement contains sentences intended by the seller. Meanwhile, the product review seems to be an honest impression of the consumer, so it is easier to trust. However, in recent years, there are some problems such as fake reviews, guerilla marketing and increase in unhelpful reviews due to increase in users. Therefore, in many e-commerce sites, each reviews have “Helpful” and “Not Helpful” buttons for consumers to evaluate the helpfulness of the review. In this paper, we consider the ranking prediction model for helpfulness of Amazon reviews. First, we redefine the helpfulness score to take its reliability into consideration. In addition, we propose a loss function that can predict the helpfulness more accurately, and describe its theoretical explanation and the results of experiments.

1　はじめに

商品レビューは，消費者が購入時の判断に参考にするため，オンラインショッピングに大きな影響を与える．広告には売手の意図した文章が含まれるのに対し，商品レビューは購入した客の正直な感想と思われるため信頼しやすい．

しかし近年，ヤラセやステマなどのレビューの悪用や利用者の増加に伴う役に立たないノイズレビューの増加により，真に有用なレビューが埋もれる問題がある．この問題に対し，主要な EC サイトでは，利用者に「このレビューは参考になりましたか?」とレビューを評価させる有用性評価が行われている．

本研究では後者の「利用者の増加に伴う役に立たないノイズレビューの増加」に焦点を当てる．既に多くのレビューが存在する場合，新たに投稿された有用なレビューが過去のものと同じように評価される可能性は低い．投稿日時に関係なくレビューそのものの有用性を推定できれば，より多くの有用な情報を消費者および出品者に提供できる．

*連絡先：九州大学大学院システム情報科学府
〒 819-0395 福岡県福岡市西区元岡 744
E-mail: t.shibata.130@s.kyushu-u.ac.jp
†連絡先：九州大学情報基盤研究開発センター
〒 819-0395 福岡県福岡市西区元岡 744
E-mail: ito.eisuke.523@m.kyushu-u.ac.jp

多くの関連研究 [1][2][3] では有用性スコアの回帰問題として取り組まれてきた．しかし利用者にとって，有用性スコアの値そのものよりも，有用なレビューほどページ上位に来ることが望まれる．本研究では有用性スコアに基づくランキング推定問題として商品レビューの有用性推定に取り組む．

本稿では，まず従来の有用性スコアの定義を見直し，スコアの信頼度を考慮できるよう再定義する．また，有用性をより正確に推定できる損失関数を提案し，その理論的説明を行う．ランキングの評価指標を用いた比較実験とその結果について述べる．

2　有用性スコア

2.1　従来の有用性スコアの定義と問題点

本研究では Amazon Review Dataset[4] を使用する．このデータセットには，各レビューに対する「役に立った」および「役に立たなかった」のボタンが押された回数が付随している．

先行研究の多くでは，各レビューの有用性を以下の式 1 で定義している．

$$Score = \frac{a}{a+b} \tag{1}$$

a :「役に立った」が押された回数

b :「役に立たなかった」が押された回数

式 1 有用性スコアは,「そのレビューを評価した人のうち何割が有用だと考えているか」を意味する．しかしこの定義には，スコアの信頼性を考慮できないという問題がある．例えば以下の 2 つのレビューにおいて，この定義による有用性スコアはどちらも 0.8 になる．

- 「役に立った」が 4 票,「役に立たなかった」が 1 票のレビュー
- 「役に立った」が 40 票,「役に立たなかった」が 10 票のレビュー

この 2 つのレビューの有用性スコア 0.8 という値は同程度に確からしいと言えるだろうか．より信頼が持てるのは後者だろう．Yang らの研究 [1][2] においても，この定義は良い有用性の推定値ではないかもしれないと指摘している．特に信頼性の低い (総票数の少ない) 場合は，この定義は教師信号というよりノイズになるとも指摘している．

本研究では，信頼性の低いレビューが 0 や 1 などの極端なスコアを取らないように，有用性スコアの定義を見直す.

2.2 有用性の確率分布

ある商品レビューに対して，ユーザは「役に立った」か「役に立たなかった」かの 2 通りで評価する．ある 1 人のユーザによる評価は式 2 のベルヌーイ分布で表現できる．

$$f(x \mid \theta) = \theta^x (1-\theta)^{1-x} \tag{2}$$

x :「役に立った」とき 1,「役に立たなかった」とき 0

$\theta : x = 1$ となる確率 (有用性スコア)

あるレビューを N 人が評価した場合，尤度は式 3 で表される．

$$L(\theta) = \prod_{i=1}^{N} f(x_i \mid \theta) = \prod_{i=1}^{N} \theta^{x_i}(1-\theta)^{1-x_i} \tag{3}$$

2.3 最尤推定値

従来の有用性スコアは最尤推定により求まる θ の値 (これを θ_{ML} とする) で定義していると考えられる．最尤推定値は尤度が最大値となる時の θ の値であり，以下のように求まる．

$$\begin{aligned}
\frac{d \log L(\theta)}{d\theta} &= \frac{d}{d\theta} \log \left(\prod_{i=1}^{N} \theta^{x_i}(1-\theta)^{1-x_i} \right) \\
&= \frac{d}{d\theta} \sum_{i=1}^{N} \left(x_i \log\theta + (1-x_i)\log(1-\theta) \right) \\
&= \sum_{i=1}^{N} \left(\frac{x_i}{\theta} - \frac{1-x_i}{1-\theta} \right) = 0 \\
\sum_{i=1}^{N} (x_i(1-\theta) - (1-x_i)\theta) &= 0 \\
\sum_{i=1}^{N} (x_i - \theta) &= 0 \\
\sum_{i=1}^{N} x_i - N\theta &= 0
\end{aligned}$$

$$\theta_{ML} = \frac{\sum_{i=1}^{N} x_i}{N} \tag{4}$$

式 4 ように，最尤推定値 θ_{ML} は，評価したユーザのうち「役に立った」と評価している人の割合を意味する．これは従来の有用性スコアの定義と等しい．2.1 節で述べたように，N の数に応じて信頼性が変わる．N が大きければ信頼でき，N が小さいときは信頼できない．式 4 は総票数 N で割られているため，θ_{ML} はその値の信頼性を考慮できない.

2.4 EAP 推定値

EAP(expected a posteriori) 推定値は，ベイズの定理より求まる事後分布の期待値である．尤度がベルヌーイ分布の場合，ベイズの定理の事前分布には，共役事前分布であるベータ分布が用いられる．事後分布の計

算を以下に示す．

$$p(\theta) = Beta(\theta \mid \alpha, \beta) = \frac{1}{B(\alpha, \beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}$$

$$\begin{aligned} f(\theta \mid x) &= \frac{f(x \mid \theta)p(\theta)}{p(x)} \\ &\propto f(x \mid \theta)p(\theta) \\ &\propto \left(\prod_{i=1}^{N} \theta^{x_i}(1-\theta)^{1-x_i}\right)\theta^{\alpha-1}(1-\theta)^{\beta-1} \\ &= \theta^{\sum_{i=1}^{N} x_i+\alpha-1}(1-\theta)^{\sum_{i=1}^{N}(1-x_i)+\beta-1} \\ &= \theta^{\alpha'-1}(1-\theta)^{\beta'-1} \\ \alpha' &= \sum_{i=1}^{N} x_i + \alpha, \beta' = \sum_{i=1}^{N}(1-x_i) + \beta \end{aligned}$$

このように，事後分布はパラメータが α', β' のベータ分布と同じ形になる．また，EAP 推定値 θ_{EAP} は，事後分布の期待値であるため，以下のようになる．

$$\theta_{EAP} = \mathbb{E}[Beta(\theta \mid \alpha', \beta')] = \frac{\alpha'}{\alpha' + \beta'}$$

$$\theta_{EAP} = \frac{\sum_{i=1}^{N} x_i + \alpha}{N + \alpha + \beta} \tag{5}$$

式??の通り，θ_{EAP} は事前分布のパラメータ α, β と観測値に基づいて値が決定する．ベータ分布は図 1 のようにパラメータ α, β により様々な形を取る．事前分布は自由に決めることが出来る．本研究では無情報事前分布である一様分布を採用する．一様分布になるベータ分布は $\alpha = 1, \beta = 1$ である．$\alpha = 1, \beta = 1$ を代入すると式??の θ_{EAP} は以下のようになる．

$$\theta_{EAP} = \frac{\sum_{i=1}^{N} x_i + 1}{N + 2} \tag{6}$$

これは，すべてのレビューに対し「役に立った」と「役に立たなかった」が 1 票ずつ入っている状態 ($Score = 0.5$) を前提としていることになる．最尤推定値 θ_{ML} と比較すると，総票数 N の少ないレビューでは θ_{EAP} は事前分布の期待値 0.5 に近い値になり，総票数の多いレビューでは θ_{EAP} は θ_{ML} に近づく．例えば以下の表 1 に示す 3 つのレビューでは，θ_{ML} は等しい．一方で，θ_{EAP} は総票数の多い方が大きな値となる．

表 1: 最尤推定値と EAP 推定値の比較

Total	Helpful	Not Helpful	θ_{ML}	θ_{EAP}
5	4	1	0.8	0.714
50	40	10	0.8	0.788
500	400	100	0.8	0.799

EAP 推定値 θ_{EAP} は，有用だと評価した人の割合 (θ_{ML}) が同じでも，その人数 N(信頼性) に応じて値が異なる．本研究では，式 6 に示す一様分布を事前分布とした EAP 推定値を有用性スコアとして定義する．

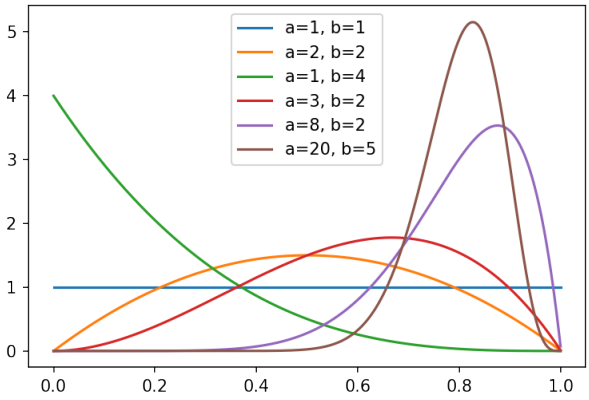

図 1: ベータ分布

3 有用性スコアの推定手法

3.1 モデル

本研究で使用するモデル全体の詳細を図 2 に示す．商品ごとに有用なレビューの特徴が異なる可能性を考慮するため，レビュー本文と商品 ID の 2 つを入力とした．単語および商品 ID はそれぞれ Embedding 層で 128 次元のベクトル w_i, e_{I_j} で表現した．レビュー本文の Encoder には 2 層の Bi-GRU を用いた．また，有用性に繋がる単語や表現に注目できるよう Self-Attention を用いた．

単語数 n の商品レビュー本文 $S = (w_1, w_2, \ldots, w_n)$ に対して，2 層の Bi-GRU を適用し隠れ状態 $H_2 = (h_{21}, h_{22}, \ldots, h_{2n}) \in \mathbb{R}^{d \times n}$ を得る．d はハイパーパラメータであり，文 Encoder が出力する隠れ状態 e_R の次元数である．本研究では $d = 1024$ とした．また，Attention の重み $A \in \mathbb{R}^{1 \times n}$ は以下の式で求まる．

$$\begin{aligned} A &= softmax(W_2 tanh(W_1 H_2)) \\ W_1 &\in \mathbb{R}^{d \times d}, W_2 \in \mathbb{R}^{1 \times d} \end{aligned}$$

Attention 層の出力 $M = (m_1, m_2, \ldots, m_n) \in \mathbb{R}^{d \times n}$ は，

$$M = A * H_2, m_i = a_i * h_{2i}$$

となり，文 Encoder の出力 e_R は

$$e_R = \sum_{i=1}^{n} m_i$$

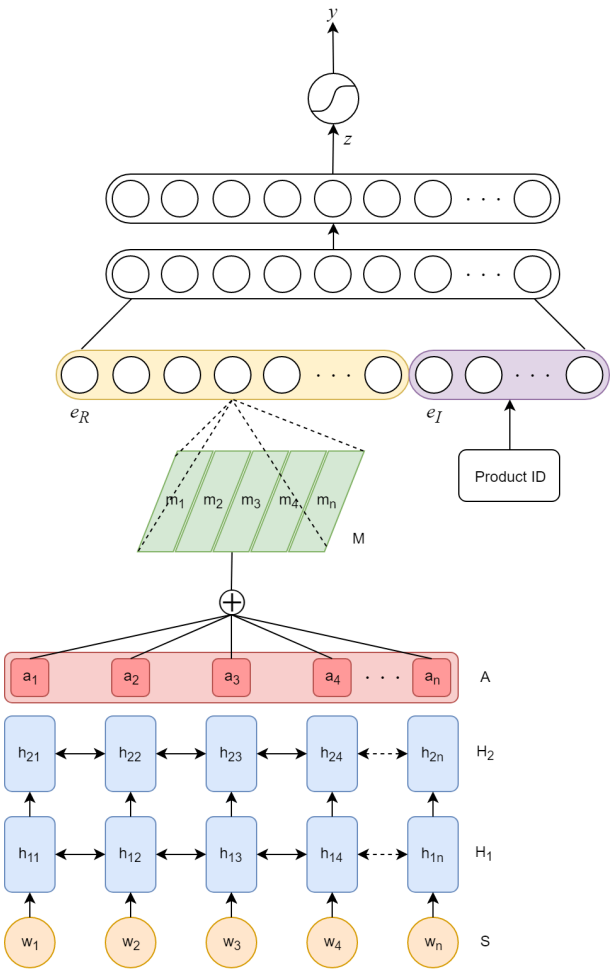

図 2: 本研究で使用するモデルの概要

となる．e_R と e_I を結合し，3 層パーセプトロンを適用することで，出力値 y を得る．出力値は $(0, 1)$ 区間上で定義されるため，シグモイド関数を用いて $y = sigmoid(z)$ とした．

$$z = MLP(e_R \oplus e_I)$$
$$y = sigmoid(z)$$

3.2 損失関数

先行研究 [1][2] では，損失関数に予測値 y と有用性スコアの平均二乗誤差 (MSE) を用いている．MSE は誤差項 ϵ に正規分布を仮定している．

$$t = y + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma)$$

しかし，有用性スコアのように，t が区間 $(0, 1)$ の有限区間で定義されている場合，誤差項 ϵ を区間 $(-\infty, \infty)$ で定義される正規分布で仮定するのは相応しくない．本研究では，t と y にシグモイド関数の逆関数であるロジット関数を適用することで，その定義域が $(-\infty, \infty)$ になるよう変換した．

$$logit(t) = logit(y) + \epsilon$$
$$\log\left(\frac{t}{1-t}\right) = z + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma)$$

有用性スコアが 0 や 1 に近いほど変換前の値より絶対値の大きな値へ変換される．極端なスコアのものほど誤差が大きく評価されるため，ランキング上位における推定性能の向上を期待する．

最終的な損失関数は $logit(t)$ と $logit(y)$ との MSE である式 7 で定義した．

$$L = \frac{1}{N}\sum_{i=1}^{N}(logit(t_i) - logit(y_i))^2 \tag{7}$$

4 実験設定と評価指標

4.1 データセット

実験で使用したデータセットについて説明する．本研究では Amazon Review Dataset のうち，表 2 に示す 4 つのカテゴリに含まれるレビューを使用した．各レビューは少なくとも 5 票以上の評価を得ており，各商品はこのようなレビューを 5 つ以上含んでいる．トークナイザに Sentencepiece[5] を使用し，語彙サイズは 8000 とした．

4.2 実験設定

本研究ではモデルの比較ではなく，損失関数の比較を行う．従来手法である予測値 y と有用性スコアの平均二乗誤差 (MSE) と，3.2 節で定義した損失関数それぞれでモデルを訓練し，テストデータに対する精度を比較する．さらに，4.2.1 節，4.2.2 節に示す 2 つのランキング推定手法，Pairwise および Listwise とも比較する．

4.2.1 Pairwise

Pairwise では 2 つのデータのスコアの大小関係を最適化する．誤差関数には主にクロスエントロピーが用いられ，どちらのデータがランキング上位であるかの分類問題を解く [7]．Kendall の順位相関係数を最適化

表 2: データセット

Category	Electronics	Home & Kitchen	CDs & Vinyl	Movies & TV
Reviews	141,985	37,895	252,566	368,453
Products	12,808	4,261	21,508	21,622
Reviews/Product	11.09	8.89	11.74	17.04
Helpfulness	0.721	0.781	0.636	0.599
# Train	127,786	34,105	227,309	331,607
# Test	14,199	3,790	25,257	36,846

し，一般に MSE などの Pointwise な手法よりも高い精度で推定できる．

$$p_{ij} = sigmoid(z_i - z_j)$$
$$t_{ij} = \begin{cases} 1 & (z_i > z_j) \\ 0.5 & (z_i = z_j) \\ 0 & (z_i < z_j) \end{cases}$$
$$L_{ij} = -t_{ij}\log(p_{ij}) - (1 - t_{ij})\log(1 - p_{ij})$$

4.2.2 Listwise

Listwise では複数のデータに対する順序関係を最適化する．一般に，Listwise な手法は Pairwise な手法に比べて，NDCG において高い精度で推定できる．Cao らの研究 [8] では，Permutation probability distribution(PPD) というデータの各並び順の起こりやすさを確率分布にしたものを定義し，真の PPD とのクロスエントロピーを損失関数 Permutation probability loss(PPL) として定義している．PPD および PPL は以下の式で定義される．π はある順序，s_i はスコアを表す．

$$P(\pi \mid \mathbf{s}) = \prod_{j}^{n} \frac{exp(s_j)}{\sum_{i=j}^{n} exp(s_i)}$$
$$PPL = -\sum P(\pi \mid \bar{\mathbf{s}}) \log P(\pi \mid \mathbf{s})$$

例えば，$n = 3, \mathbf{s} = (s_1, s_2, s_3), \pi = \langle 1, 2, 3 \rangle$ に対する PPD は，

$$P(\pi) = \frac{exp(s_1)}{exp(s_1) + exp(s_2) + exp(s_3)} \frac{exp(s_2)}{exp(s_2) + exp(s_3)} \frac{exp(s_3)}{exp(s_3)}$$

$n = 3, \mathbf{s} = (s_1, s_2, s_3), \pi' = \langle 3, 2, 1 \rangle$ に対する PPD は，

$$P(\pi') = \frac{exp(s_3)}{exp(s_1) + exp(s_2) + exp(s_3)} \frac{exp(s_2)}{exp(s_2) + exp(s_1)} \frac{exp(s_1)}{exp(s_1)}$$

となる．

PPL の計算にはすべての並び順に対する $n!$ 通りの PPD を計算する必要がある．計算量を削減するため，上位 k 個の並び順だけを考慮し，$n!/(n-k)!$ の計算量で求まる Top-k PPL が提案されている．特に $k = 1$ のとき，PPL は softmax で表現できる．

$$P(\pi \mid \mathbf{s}) = \prod_{j}^{k} \frac{exp(s_j)}{\sum_{i=j}^{n} exp(s_i)}$$

本研究では Top-1 PPL を Listwise な損失関数として採用した．

4.2.3 学習時の設定

パラメータの最適化には MomentumSGD を用いた．また，Cyclic Cosine Annealing により学習率のスケジューリングを行い，40epoch/cycle×5 の計 200epoch 訓練した．さらに各周期の最後にモデルを保存し，推論時にアンサンブルを行う Snapshot Ensembles[6] を適用した．

4.3 評価指標

評価指標には Kendall の順位相関係数と Normalized Discounted Cumulative Gain(NDCG) を用いた．これらはランキング学習の評価指標として用いられる．一般に回帰分析では MSE や相関係数，決定係数 R^2 などが用いられるが，対象とする商品レビューでは有用なレビューほどページ上位に来るよう予測できることが望まれる．特に，ユーザの目につきやすいランキング上位のレビューほど，その順位を正確に予測する必要がある．NDCG の算出時には，テストデータ全体に対する値とランキング上位 1%に対する値を計算した．

5 実験結果と考察

表 3 に実験結果を示す．従来の損失関数を MSE，本研究の提案手法を Logit とした．

従来の損失関数である MSE と提案手法を比較すると，すべてのカテゴリおよび評価指標で提案手法の方が良い結果となった．Pairwise，Listwise な手法と比較しても，

表 3: 各カテゴリにおける推定精度

	Electronics				Home & Kitchen				CDs & Vinyl				Movies & TV			
Lossfunc	MSE	Logit	Pair	List	MSE	Logit	Pair	List	MSE	Logit	Pair	List	MSE	Logit	Pair	List
Kendall	0.339	0.413	**0.414**	0.399	0.160	0.318	**0.321**	0.267	0.427	**0.505**	0.491	0.470	0.402	**0.516**	0.475	0.437
NDCG@1%	0.867	**0.909**	**0.909**	0.902	0.836	**0.891**	0.850	0.851	0.842	**0.893**	0.887	0.890	0.819	**0.888**	0.868	0.875
NDCG@All	0.984	**0.988**	**0.988**	0.987	0.977	**0.985**	0.984	0.982	0.983	**0.987**	0.987	0.987	0.980	**0.986**	0.985	0.985

提案手法の方がおおむね優れている．カテゴリ”Electronics”と”Home&Kitchen”においては，Kendall による評価で Pairwise が最も良い結果となった．Pairwise では，スコアの大きさに応じ損失に重みを加える手法もあり，スコア上位のものにより大きな重みを掛けることで，NGCG@1%において性能向上の可能性がある．

また，一般に Listwise は Pairwise よりも優れているとされるが，今回の実験では Pairwise よりも悪い結果となった．Listwise については，Top-k PPL の k の値 (今回は $k = 1$) によって結果が変わるため，今後，値を変えて比較する必要がある．

6 おわりに

本研究では Amazon の商品レビューを対象に，レビューの有用性に基づくランキングを推定した．有用性スコアを EAP 推定値で定義することで，信頼性を考慮できるようにした．多くの関連研究で用いられてきた損失関数の問題点を指摘し，ロジット変換を適用した MSE を最小化することで，より高いランキング推定精度が得られることを示した．

ランキング学習に用いられる Pairwise や Listwise な手法も試した．しかし，ランキング上位の推定性能において良い結果が得られなかった．これらは上位のデータに対し重み付けを行うなどの工夫で性能が向上する可能性があり，今後，比較実験を行いたい．

また，本研究ではモデルによる比較をしていない．より良いランキング推定が出来るよう，モデルの改良も行いたい．

参考文献

[1] Yinfei Yang, Minghui Qiu, Yaowei Yan and Forrest Sheng Bao: Semantic analysis and helpfulness prediction of text for online product reviews, ACL-IJCNLP, Volume 2, Pages 38-44, 2015.

[2] Yinfei Yang, Cen Chen and Forrest Sheng Bao: Aspect-Based Helpfulness Prediction for Online Product Reviews, IEEE 28th International Conference on Tools with Artificial Intelligence, 2016.

[3] Cen Chen, Yinfei Yang, Jun Zhou, Xiaolong Li, and Forrest Sheng Bao: Cross-Domain Review Helpfulness Prediction Based on Convolutional Neural Networks with Auxiliary Domain Discriminators, In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL),Volume 2 (Short Papers), Association for Computational Linguistics, 602–607, 2018.

[4] R. He, J. McAuley: Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering, WWW, 2016.

[5] Kudo Taku and Richardson John: SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing, In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018.

[6] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger: Snapshot ensembles: Train 1, get M for free, ICLR submission, 2017.

[7] Chris J.C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender: Learning to Rank using Gradient Descent, ICML ’05 Proceedings of the 22nd international conference on Machine learning, Pages 89-96, 2005.

[8] Z. Cao, T. Qin, T.- Y. Liu, M.-F. Tsai and H. Li: Learning to Rank: from Pairwise Approach to Listwise Approach, Proceedings of the 24th international conference on Machine learning, 2007.