Title: Learning to rank

URL: https://en.wikipedia.org/wiki/Learning_to_rank

PageID: 25050663

Categories: Category:Information retrieval techniques, Category:Machine learning, Category:Ranking functions

-----

Supervised learning

Unsupervised learning

Semi-supervised learning

Self-supervised learning

Reinforcement learning

Meta-learning

Online learning

Batch learning

Curriculum learning

Rule-based learning

Neuro-symbolic AI

Neuromorphic engineering

Quantum machine learning

Classification

Generative modeling

Regression

Clustering

Dimensionality reduction

Density estimation

Anomaly detection

Data cleaning

AutoML

Association rules

Semantic analysis

Structured prediction

Feature engineering

Feature learning

Learning to rank

Grammar induction

Ontology learning

Multimodal learning

Apprenticeship learning

Decision trees

Ensembles Bagging Boosting Random forest

Bagging

Boosting

Random forest

k -NN

Linear regression

Naive Bayes

Artificial neural networks

Logistic regression

Perceptron

Relevance vector machine (RVM)

Support vector machine (SVM)

BIRCH

CURE

Hierarchical

k -means

Fuzzy

Expectation–maximization (EM)

DBSCAN

OPTICS

Mean shift

Factor analysis

CCA

ICA

LDA

NMF

PCA

PGD

t-SNE

SDL

Graphical models Bayes net Conditional random field Hidden Markov

Bayes net

Conditional random field

Hidden Markov

RANSAC

k -NN

Local outlier factor
Isolation forest
Autoencoder
Deep learning
Feedforward neural network
Recurrent neural network LSTM GRU ESN reservoir computing
LSTM
GRU
ESN
reservoir computing
Boltzmann machine Restricted
Restricted
GAN
Diffusion model
SOM
Convolutional neural network U-Net LeNet AlexNet DeepDream
U-Net
LeNet
AlexNet
DeepDream
Neural field Neural radiance field Physics-informed neural networks
Neural radiance field
Physics-informed neural networks
Transformer Vision
Vision
Mamba
Spiking neural network
Memtransistor
Electrochemical RAM (ECRAM)
Q-learning
Policy gradient
SARSA
Temporal difference (TD)
Multi-agent Self-play
Self-play
Active learning
Crowdsourcing
Human-in-the-loop

Mechanistic interpretability

RLHF

Coefficient of determination

Confusion matrix

Learning curve

ROC curve

Kernel machines

Bias–variance tradeoff

Computational learning theory

Empirical risk minimization

Occam learning

PAC learning

Statistical learning

VC theory

Topological deep learning

AAAI

ECML PKDD

NeurIPS

ICML

ICLR

IJCAI

ML

JMLR

Glossary of artificial intelligence

List of datasets for machine-learning research List of datasets in computer vision and image processing

List of datasets in computer vision and image processing

Outline of machine learning

v

t

e

Learning to rank [1] (LTR) or machine-learned ranking (MLR) is the application of machine learning , often supervised , semi-supervised or reinforcement learning , in the construction of ranking models for information retrieval and recommender systems. [2] Training data may, for example, consist of lists of items with some partial order specified between items in each list. This order is typically induced by giving a numerical or ordinal score or a binary judgment (e.g. "relevant" or "not relevant") for each item. The goal of constructing the ranking model is to rank new, unseen lists in a similar way to rankings in the training data.

Applications

In information retrieval

Ranking is a central part of many information retrieval problems, such as document retrieval , collaborative filtering , sentiment analysis , and online advertising .

A possible architecture of a machine-learned search engine is shown in the accompanying figure.

Training data consists of queries and documents matching them together with the relevance degree of each match. It may be prepared manually by human assessors (or raters , as Google calls them), who check results for some queries and determine relevance of each result. It is not feasible to check the relevance of all documents, and so typically a technique called pooling is used — only the top few documents, retrieved by some existing ranking models are checked. This technique may introduce selection bias. Alternatively, training data may be derived automatically by analyzing clickthrough logs (i.e. search results which got clicks from users), [3] query chains , [4] or such search engines' features as Google's (since-replaced) SearchWiki . Clickthrough logs can be biased by the tendency of users to click on the top search results on the assumption that they are already well-ranked.

Training data is used by a learning algorithm to produce a ranking model which computes the relevance of documents for actual queries.

Typically, users expect a search query to complete in a short time (such as a few hundred milliseconds for web search), which makes it impossible to evaluate a complex ranking model on each document in the corpus, and so a two-phase scheme is used. [5] First, a small number of potentially relevant documents are identified using simpler retrieval models which permit fast query evaluation, such as the vector space model , Boolean model , weighted AND, [6] or BM25 . This phase is called top- k {\displaystyle k} document retrieval and many heuristics were proposed in the literature to accelerate it, such as using a document's static quality score and tiered indexes. [7] In the second phase, a more accurate but computationally expensive machine-learned model is used to re-rank these documents.

In other areas

Learning to rank algorithms have been applied in areas other than information retrieval:

In machine translation for ranking a set of hypothesized translations; [8]

In computational biology for ranking candidate 3-D structures in protein structure prediction problems; [8]

In recommender systems for identifying a ranked list of related news articles to recommend to a user after he or she has read a current news article. [9]

Feature vectors

For the convenience of MLR algorithms, query-document pairs are usually represented by numerical vectors, which are called feature vectors . Such an approach is sometimes called bag of features and is analogous to the bag of words model and vector space model used in information retrieval for representation of documents.

Components of such vectors are called features , factors or ranking signals . They may be divided into three groups (features from document retrieval are shown as examples):

Query-independent or static features — those features, which depend only on the document, but not on the query. For example, PageRank or document's length. Such features can be precomputed in off-line mode during indexing. They may be used to compute document's static quality score (or static rank), which is often used to speed up search query evaluation. [7] [10]

Query-dependent or dynamic features — those features, which depend both on the contents of the document and the query, such as TF-IDF score or other non-machine-learned ranking functions.

Query-level features or query features , which depend only on the query. For example, the number of words in a query.

Some examples of features, which were used in the well-known LETOR dataset:

TF, TF-IDF , BM25 , and language modeling scores of document's zones (title, body, anchors text, URL) for a given query;

Lengths and IDF sums of document's zones;

Document's PageRank , HITS ranks and their variants.

Selecting and designing good features is an important area in machine learning, which is called feature engineering .

Evaluation measures

There are several measures (metrics) which are commonly used to judge how well an algorithm is doing on training data and to compare the performance of different MLR algorithms. Often a learning-to-rank problem is reformulated as an optimization problem with respect to one of these metrics.

Examples of ranking quality measures:

Mean average precision (MAP);

DCG and NDCG ;

Precision @ n , NDCG@ n , where "@ n " denotes that the metrics are evaluated only on top n documents;

Mean reciprocal rank ;

Kendall's tau ;

Spearman's rho .

DCG and its normalized variant NDCG are usually preferred in academic research when multiple levels of relevance are used. [11] Other metrics such as MAP, MRR and precision, are defined only for binary judgments.

Recently, there have been proposed several new evaluation metrics which claim to model user's satisfaction with search results better than the DCG metric:

Expected reciprocal rank (ERR); [12]

Yandex 's pfound. [13]

Both of these metrics are based on the assumption that the user is more likely to stop looking at search results after examining a more relevant document, than after a less relevant document.

Approaches

Learning to Rank approaches are often categorized using one of three approaches: pointwise (where individual documents are ranked), pairwise (where pairs of documents are ranked into a relative order), and listwise (where an entire list of documents are ordered).

Tie-Yan Liu of Microsoft Research Asia has analyzed existing algorithms for learning to rank problems in his book Learning to Rank for Information Retrieval . [1] He categorized them into three groups by their input spaces, output spaces, hypothesis spaces (the core function of the model) and loss functions : the pointwise, pairwise, and listwise approach. In practice, listwise approaches often outperform pairwise approaches and pointwise approaches. This statement was further supported by a large scale experiment on the performance of different learning-to-rank methods on a large collection of benchmark data sets. [14]

In this section, without further notice, x {\displaystyle x} denotes an object to be evaluated, for example, a document or an image, f (x) {\displaystyle f(x)} denotes a single-value hypothesis, h (·) {\displaystyle h(\cdot)} denotes a bi-variate or multi-variate function and L (·) {\displaystyle L(\cdot)} denotes the loss function.

Pointwise approach

In this case, it is assumed that each query-document pair in the training data has a numerical or ordinal score. Then the learning-to-rank problem can be approximated by a regression problem — given a single query-document pair, predict its score. Formally speaking, the pointwise approach aims at learning a function f (x) {\displaystyle f(x)} predicting the real-value or ordinal score of a document x {\displaystyle x} using the loss function L (f ; x j , y j) {\displaystyle L(f;x_{j},y_{j})} .

A number of existing supervised machine learning algorithms can be readily used for this purpose. Ordinal regression and classification algorithms can also be used in pointwise approach when they are used to predict the score of a single query-document pair, and it takes a small, finite number of values.

Pairwise approach

In this case, the learning-to-rank problem is approximated by a classification problem — learning a binary classifier h (x u , x v) {\displaystyle h(x_{u},x_{v})} that can tell which document is better in a given pair of documents. The classifier shall take two documents as its input and the goal is to minimize a loss function L (h ; x u , x v , y u , v) {\displaystyle L(h;x_{u},x_{v},y_{u,v})} . The loss function typically reflects the number and magnitude of inversions in the induced ranking.

In many cases, the binary classifier h (x u , x v) {\displaystyle h(x_{u},x_{v})} is implemented with a scoring function f (x) {\displaystyle f(x)} . As an example, RankNet [15] adapts a probability model and defines h (x u , x v) {\displaystyle h(x_{u},x_{v})} as the estimated probability of the document x u {\displaystyle x_{u}} has higher quality than x v {\displaystyle x_{v}} :

where CDF (·) {\displaystyle {\text{CDF}}(\cdot)} is a cumulative distribution function , for example, the standard logistic CDF , i.e.

Listwise approach

These algorithms try to directly optimize the value of one of the above evaluation measures, averaged over all queries in the training data. This is often difficult in practice because most evaluation measures are not continuous functions with respect to ranking model's parameters, and so continuous approximations or bounds on evaluation measures have to be used. For example the SoftRank algorithm. [16] LambdaMART is a pairwise algorithm which has been empirically shown to approximate listwise objective functions. [17]

List of methods

A partial list of published learning-to-rank algorithms is shown below with years of first publication of each method:

Regularized least-squares based ranking. The work is extended in [26] to learning to rank from general preference graphs.

Note: as most supervised learning-to-rank algorithms can be applied to pointwise, pairwise and listwise case, only those methods which are specifically designed with ranking in mind are shown above.

History

Norbert Fuhr introduced the general idea of MLR in 1992, describing learning approaches in information retrieval as a generalization of parameter estimation; [49] a specific variant of this approach (using polynomial regression) had been published by him three years earlier. [18] Bill Cooper proposed logistic regression for the same purpose in 1992 [19] and used it with his Berkeley research group to train a successful ranking function for TREC . Manning et al. [50] suggest that these early works achieved limited results in their time due to little available training data and poor machine learning techniques.

Several conferences, such as NeurIPS , SIGIR and ICML have had workshops devoted to the learning-to-rank problem since the mid-2000s (decade).

Practical usage by search engines

Commercial web search engines began using machine-learned ranking systems since the 2000s (decade). One of the first search engines to start using it was AltaVista (later its technology was acquired by Overture , and then Yahoo), which launched a gradient boosting -trained ranking function in April 2003. [51] [52]

Bing 's search is said to be powered by RankNet algorithm, [53] [when?] which was invented at Microsoft Research in 2005.

In November 2009 a Russian search engine Yandex announced [54] that it had significantly increased its search quality due to deployment of a new proprietary MatrixNet algorithm, a variant of gradient boosting method which uses oblivious decision trees. [55] Recently they have also sponsored a machine-learned ranking competition "Internet Mathematics 2009" [56] based on their own search engine's production data. Yahoo has announced a similar competition in 2010. [57]

As of 2008, Google 's Peter Norvig denied that their search engine exclusively relies on machine-learned ranking. [58] Cuil 's CEO, Tom Costello, suggests that they prefer hand-built models because they can outperform machine-learned models when measured against metrics like click-through rate or time on landing page, which is because machine-learned models "learn what people say they like, not what people actually like". [59]

In January 2017, the technology was included in the open source search engine Apache Solr . [60] It is also available in the open source OpenSearch and Elasticsearch . [61] [62] These implementations make learning to rank widely accessible for enterprise search.

Vulnerabilities

Similar to recognition applications in computer vision , recent neural network based ranking algorithms are also found to be susceptible to covert adversarial attacks , both on the candidates and the queries. [63] With small perturbations imperceptible to human beings, ranking order could be arbitrarily altered. In addition, model-agnostic transferable adversarial examples are found to be possible, which enables black-box adversarial attacks on deep ranking systems without requiring access to their underlying implementations. [63] [64]

Conversely, the robustness of such ranking systems can be improved via adversarial defenses such as the Madry defense. [65]

See also

Content-based image retrieval

Multimedia information retrieval

Image retrieval

Triplet loss

References

External links

LETOR: A Benchmark Collection for Research on Learning to Rank for Information Retrieval

Yandex's Internet Mathematics 2009

Yahoo! Learning to Rank Challenge

Microsoft Learning to Rank Datasets