A Unified Energy-based Framework for Learning to Rank

Yi Fang
Department of Computer Engineering
Santa Clara University
Santa Clara 95053, CA, USA
yfang@scu.edu

Mengwen Liu
College of Computing and Informatics
Drexel University
Philadelphia 19104, PA, USA
ml943@drexel.edu

ABSTRACT

Learning to Rank (L2R) has emerged as one of the core machine learning techniques for IR. On the other hand, Energy-Based Models (EBMs) capture dependencies between variables by associating a scalar energy to each configuration of the variables. They have produced impressive results in many computer vision and speech recognition tasks. In this paper, we introduce a unified view of Learning to Rank that integrates various L2R approaches in an energy-based ranking framework. In this framework, an energy function associates low energies to desired documents and high energies to undesired results. Learning is essentially the process of shaping the energy surface so that desired documents have lower energies. The proposed framework yields new insights into learning to rank. First, we show how various existing L2R models (pointwise, pairwise, and listwise) can be cast in the energy-based framework. Second, new L2R models can be constructed based on existing EBMs. Furthermore, inspired by the intuitive learning process of EBMs, we can devise novel energy-based models for ranking tasks. We introduce several new energy-based ranking models based on the proposed framework. The experiments are conducted on the public LETOR 4.0 benchmarks and demonstrate the effectiveness of the proposed models.

Keywords

Learning to Rank; Energy-based Models

1. INTRODUCTION

Ranking is the central problem in many IR tasks including document retrieval, entity search, question answering, meta-search, collaborative filtering, online advertisement, and so on. These tasks usually work with high dimensional feature vector representations of the items to be ranked. The typical features range from query independent ones to information measuring the match between user or query and retrieved item. The dimensionality of feature vectors and the complexity of statistical relationships involved are such that accurate results cannot be achieved by designing the relevant ranking functions manually. Therefore, learning to rank (L2R) from examples has become the dominant approach for designing and optimizing ranking systems. Recent years have witnessed significant efforts on research and development of learning to rank technologies. L2R models can be classified into three broad families: pointwise, pairwise, and listwise methods [16, 15]. Benchmark datasets like LETOR [17] have been released to facilitate the research on learning to rank. It has become a key technology in the industry. Several major search engine companies are using L2R techniques to train their ranking models [16].

On the other hand, energy-based models (EBMs) [13, 12] are a family of learning models that capture dependencies between variables by associating a scalar energy to each configuration of the variables. Making a decision (an inference) with an EBM consists of comparing the energies associated with various configurations of the variable to be predicted, and choosing the one with the lowest energy. Such systems are trained to associate low energies to the desired configurations and higher energies to undesired ones. Unlike probabilistic models that associate a probability to those configurations, energy-based models eliminate the need for proper normalization of probability distributions. The main question in EBMs is how to design a loss function so that minimizing this loss function with respect to the parameter vector will have the effect of "digging holes and building hills" at the required places on the energy surface. Energy-based models have been widely applied to computer vision and speech recognition tasks, and demonstrated their effectiveness and efficiency [20, 18]. Some recent successes of deep learning architectures are largely due to energy-based learning [25].

In this paper, we attempt to shed new light on learning to rank by formulating it in an energy-based framework. To the best of our knowledge, no prior work has investigated the link between learning to rank and energy-based learning. We present a unified energy-based framework for learning to rank. We demonstrate how various existing learning to rank models (pointwise, pairwise, and listwise) can be cast in this framework. Moreover, we propose new learning to rank techniques based on the energy-based framework. The advantages of the energy-based framework for learning to rank are multi-fold. First of all, it can demonstrate the similarity and difference between various L2R methods from the energy-based perspective, which may help us gain further insights into the strengths and weaknesses of each ranking algorithm. Moreover, there exists an extensive research on energy-based models in machine learning and computer vision communities. We can explore and apply them to learning to rank problems. Last but not the least, the energy-based framework can provide sensible guidelines to design and propose new learning to rank models. The energy-based learning aims to reshape the energy functions so that the desired outcomes have

ICTIR '16, September 12-16, 2016, Newark, DE, USA

DOI: http://dx.doi.org/10.1145/2970398.2970416

lower energies. The loss and energy functions can be intuitively designed to achieve this effect.

2. RELATED WORK

2.1 Learning to Rank

Learning to rank models can be classified into three broad families: pointwise, pairwise, and listwise methods [16]. Pointwise approaches postulate a scoring function and attempt to estimate the relevance score of every item. The relevance score is typically the rank of the item in the list or a transformed version of it. At prediction time, the items returned for a query are sorted according to their estimated scores. Linear and logistic regression are examples of scoring functions used in pointwise approaches. Pairwise methods score ordered pairs of items instead of individual items. The goal is now to learn the order of such pairs correctly. In other words, the task is to score more relevant items higher than less relevant ones. In general, this approach is preferable to the pointwise approach, because it does not require to learn absolute relevance scores. RankSVM [10] is one of the most popular pairwise approaches. It formalizes ranking as a binary classification problem of item pairs and uses support vector machines as the underlying binary classifier. RankBoost [8] is another pairwise ranking model, where boosting is used to learn the ranking. The idea is to construct a sequence of weak rankers over iteratively reweighted training data, and then to make rank predictions using a linear combination of the weak learners. While the predictive power of RankBoost is greater in theory, it only marginally improves the quality of ranking in practice. Burges et al. [3] propose the RankNet algorithm, which is also based on pairwise classification like RankSVM and RankBoost. The major difference lies in that it employs Neural Network as ranking model and uses cross entropy as loss function. Finally, listwise approaches assume that the training examples are lists of ranked items. They attempt to minimize a loss function defined over the whole list instead of ordered pairs extracted from the list. ListMLE [29] and ListNet [4] are two representative listwise models. The loss functions are defined using the probability distribution on permutations. AdaRank [30] is another listwise approach, based instead on boosting. There exist an abundance of learning to rank techniques in the literature. Liu [16] and Li [15] provide two comprehensive surveys.

2.2 Energy-based Learning

The energy-based framework was first proposed by LeCun and Huang [13, 11] as a deterministic alternative to probabilistic graphical models. It provides a very general framework for dealing with learning systems, and immediately puts machine learning to the scope of mathematical optimization. Zhang [31] proves that probably approximately correct (PAC) learning is guaranteed for the energy-based learning. Energy-based models were successfully applied to various machine learning tasks including computer vision [20], speech recognition [18], unsupervised learning [23], reinforcement learning [9], relational learning [1], missing value imputation [2]. BoltzRank [27] is the only explicit energy-based L2R model in the literature, based on an energy function that depends on a scoring function composed of individual and pairwise potentials. To the best of our knowledge, no prior work has systematically studied the relationship between learning to rank and energy-based learning.

Recently, energy-based models are used to learn deep, distributed representations of high-dimensional data (such as images) and model high-order dependencies. An important class of energy-based models are Restricted Boltzmann Machines [28]. Energy-based deep learning models have been applied to a range of challenging tasks including motion capture modeling [26], modeling of transformations in natural images [19], and visual tracking [14]. Establishing the link between L2R and EBMs may facilitate applications of deep learning to information retrieval.

Table 1: Some commonly used energy functions in Energy-Based Models [11].

$\frac{1}{2}\|\|f_\theta(x) - y\|\|^2$
$\|\|f_\theta(x) - y\|\|_1$
$-yf_\theta(x)$
$\frac{1}{2}\|\|f_{\theta_x}(x) - g_{\theta_y}(y)\|\|^2$
$\sum_{k=1}^{K} \delta(y-k)\|\|U^k - f_\theta(x)\|\|^2$

Table 2: A list of commonly used loss functions in Energy-Based Models [11].

Energy loss	$E_\theta(x, y)$
Perceptron	$E_\theta(x, y) - \min_{y \in Y} E_\theta(x, y)$
Hinge	$\max(0, m + E_\theta(x, y) - E_\theta(x, \bar{y}))$
Log	$\log(1 + e^{E_\theta(x,y) - E_\theta(x,\bar{y})})$
MCE	$(1 + e^{-(E_\theta(x,y) - E_\theta(x,\bar{y}))})^{-1}$
LVQ2	$\min(M, \max(0, E_\theta(x, y) - E_\theta(x, \bar{y})))$
square-square	$E_\theta(x, y)^2 - (\max(0, m - E_\theta(x, \bar{y})))^2$
square-exp	$E_\theta(x, y)^2 + \beta e^{-E_\theta(x,\bar{y})}$
NLL	$E_\theta(x, y) + \frac{1}{\beta} \log \sum_{y \in Y} e^{-\beta E_\theta(x,y)}$
MEE	$1 - \frac{e^{-\beta E_\theta(x,y)}}{\sum_{y \in Y} e^{-\beta E_\theta(x,y)}}$

3. BACKGROUND

3.1 Energy-based Models

The entire framework of energy-based models, by its name, is centered around the concept of *energy*. It captures dependencies by associating a scalar energy (a measure of compatibility) $E(x, y)$ to each configuration of the input variable x and output y. In inference, i.e., making prediction or decision, the model produces the answer $y \in Y$ that is most compatible with the observed x, for which $E(x, y)$ is the smallest:

$$y^* = \underset{y \in Y}{\operatorname{argmin}} E(x, y) \tag{1}$$

As a result, model learning consists in finding an energy function that associates low energies to correct values of the variables, and higher energies to incorrect values. The energy function is often assumed within a family of energy functions E_θ indexed by parameter θ. The energy function could be as simple as a linear combination of basis functions or a set of neural network architectures with weight values. One advantage of the energy-based framework is that it puts very little restrictions on the nature of the architecture of the energy function. Table 1 contains some common energy functions.

A *loss functional* (function of function) is defined over energy functions. It is minimized during learning and used to measure the quality of the available energy functions. Within this common inference/learning framework, the wide choices of energy functions and loss functionals allow for the design of many types of learning models, both probabilistic and non-probabilistic. Table 2 shows a list of commonly used loss functions in EBMs. In the table, $\bar{y}$ denotes the most offending incorrect answers, i.e., the answer that has the lowest energy among all the incorrect answers.

EBMs have several advantages over maximum likelihood learning. By trying to model the whole joint distribution of a data set, a large part of the flexibility of probabilistic models is used to capture relationships that might not be necessary for the task of interest. An energy model with a deterministic inference method can make predictions that are directly optimized for the task of interest itself. Moreover, since the normalization constant of many generative models is intractable, inference needs to be done with methods like sampling or variational inference. Deterministic energy-based models circumvent this problem. LeCun et al. [11] provides a survey on energy-based models.

3.2 Notations

Learning to rank is comprised of training and testing as a supervised learning task. The training data contains queries, documents, and relevance judgments. Each query is associated with a number of documents. The relevance of the documents with respect to the query is represented by a label which is at multiple grades. The higher grade a document has, the more relevant the document is. Suppose that Q is the query set, D is the document set, and $R = \{1, 2, ..., l\}$ is the label set. There exists a total order between the grades $l \succ l-1 ... \succ 1$, where $\succ$ denotes the order relation. Further suppose that $\{q_1, q_2, ..., q_m\}$ is the set of queries for training and q_i is the i-th query. $D_i = \{d_{i,1}, d_{i,2}, ..., d_{i,n_i}\}$ is the set of documents associated with query q_i and $r_i = \{r_{i,1}, r_{i,2}, ..., r_{i,n_i}\}$ denotes the corresponding labels of those documents, where n_i denotes the sizes of D_i; $d_{i,j}$ denotes the j-th document in D_i; and $r_{i,j} \in R$ represents the relevance degree of $d_{i,j}$ with respect to q_i. A feature vector $\mathbf{x_{i,j}}$ is created from each query-document pair $(q_i, d_{i,j})$.

4. AN ENERGY-BASED RANKING FRAMEWORK

The main goal of learning to rank is to learn the relationship between document d and its relevance r given query q. We propose an energy-based ranking framework to encode dependencies among them, by associating a scalar energy $E_\theta(q, d, r)$ to each configuration of d and r given q. The family of possible energy functions is parameterized by a parameter vector θ, which is to be learned from the training data. Similar to other EBMs, this energy function can be viewed as a measure of "compatibility" between d and r given q. In the following sections, we use the convention that small energy values correspond to highly compatible configurations of the variables, while large energy values correspond to highly incompatible configurations of the variables. There are three key components in the energy-based ranking framework.

- *Loss functional* is used to measure the quality of the available energy functions. Unlike the traditional loss functions in machine learning, the loss function in energy-based learning is defined on energy functions and thus is a *functional*. Similar to the other loss functions, it is minimized during the training process.
- *Learning* consists in finding an energy function that associates low energies with the desired documents, and higher energies with the undesired documents. It is worth noting that the existing EBMs only adjusts the energies on the same instance, but for different output values (correct vs. incorrect). For ranking problems, we attempt to adjust the energies on different instances (i.e., documents), especially for pairwise and listwise learning. This is one of the major differences between the existing EBMs and the proposed ranking framework.
- *Ranking* generates a list of documents that are ranked based on their energies in the ascending order.

Mathematically, to train an energy-based ranking model, we minimize the loss functional with respect to θ as follows

$$\theta^* = \min_\theta \Big(\frac{1}{m} \sum_{i=1}^{m} L_q\big(E_\theta(q_i, d, r)\big) + R(\theta) \Big) \tag{2}$$

where $L_q\big(E_\theta(q_i, d, r)\big)$ is the per-query loss functional defined on the energy function $E_\theta(q_i, d, r)$. $R(\theta)$ is the regularizer and can be used to embed our prior knowledge about which energy functions in our family are preferable to others.

The ranking process consists of two steps in general. The first step is to find the relevance degree r that is most compatible with the document d given query q_i and model parameter θ (learned from training), which is to minimize the energy function with respect to r:

$$E_\theta(q_i, d) = \min_{r \in R} E_\theta(q_i, d, r) \tag{3}$$

The documents can then be ranked based on $E_\theta(q_i, d)$ in ascending order. In other words, the most relevant document given query q_i is

$$d^* = \operatorname*{argmin}_{d \in D_i} E_\theta(q_i, d) = \operatorname*{argmin}_{d \in D_i} \min_{r \in R} E_\theta(q_i, d, r) \tag{4}$$

In the cases where the energy function does not depend on r, we can just rank the documents based on $E_\theta(q_i, d)$. It is worth noting that the energy function is minimized during the ranking process while the loss functional is minimized during the learning process.

Besides the advantages of EBMs pointed out in Section 3.1, the key characteristic of energy-based learning is the process of reshaping the energy function based on training data so that the desired results would have lower energies. It can be viewed that the loss function is operated on energy functions instead of parameters. This functional point of view can shed new light on learning to rank. With a properly designed loss function, the energy-based ranking process should have the effect of "pushing down" on the energies of the desired documents and "pulling up" on the undesired ones. The following subsections will cast several existing learning to rank models in the energy-based ranking framework. In Section 5, we derive novel learning to rank models based on this framework.

4.1 Pointwise

In the pointwise approach, the ranking problem is transformed to classification or regression. The existing methods for classification or regression are applied. The loss function in learning is pointwise in the sense that it is defined on a single object (feature vector). The energy-based models have been studied extensively for traditional classification and regression models. For completeness, we just briefly show how some widely used classification models including support vector machine (SVM), logistic regression, and linear regression can be cast in the energy-based ranking framework.

For simplicity, assuming the relevance is binary: $r_{ij} \in \{1, -1\}$, the energy function can be defined as:

$$E_\theta(q_i, d_{ij}, r_{ij}) = -r_{ij} f(q_i, d_{ij}; \theta) = -r_{ij} \theta^{\mathrm{T}} \mathbf{x_{i,j}} \tag{5}$$

where $f(q_i, d_{ij}; \theta)$ is a discriminant function parameterized by θ and assumed a linear model here. By plugging this energy function into the hinge loss in Table 2, we obtain the per-query loss function as follows

$$L_q = \sum_{j=1}^{D_i} \max(0, M + 2 r_{ij} \theta^{\mathrm{T}} \mathbf{x_{i,j}}) \tag{6}$$

where M is the margin parameter. If the regularizer takes the form $||\theta||_2^2$, the loss will result in the linear SVM.

If we plug the energy function in Eqn.(5) into the Log loss in Table 2, the per-query loss function becomes:

$$L_q = \sum_{j=1}^{D_i} \log\left(1 + \exp(-2r_{ij}\theta^{\mathrm{T}}\mathbf{x_{i,j}})\right) \quad (7)$$

which gives the logistic regression model. The loss functions in Eqn.(6) and Eqn.(7) are slightly different from the standard ones for SVM and logistic regression, but they are equivalent since the multiplier of 2 can be absorbed into the parameter θ.

If an energy function is defined as the squared error between $\theta^{\mathrm{T}}\mathbf{x_{i,j}}$ and r_{ij} as follows:

$$E_\theta(q_i, d_{ij}, r_{ij}) = (\theta^{\mathrm{T}}\mathbf{x_{i,j}} - r_{ij})^2 \quad (8)$$

then the Energy loss, Perceptron loss, and negative log-likelihood (NLL) loss in Table 2 are all equivalent and lead to the regression loss function in the pointwise model called Subset Ranking with Regression [6]. The reason is the contrastive term of the NLL loss becomes constant since it is a Gaussian integral with a constant variance, and that of the Perceptron loss is zero.

4.2 Pairwise

The pairwise approach does not focus on accurately predicting the relevance degree of each document; instead, it cares about the relative order between two documents. In this sense, it is closer to the concept of "ranking" than the pointwise approach. In this section, we cast two representative pairwise L2R models, RankSVM and RankNet, in the energy-based ranking framework.

4.2.1 RankSVM

RankSVM [10] is one of the first learning to rank methods. It is based on the pairwise comparison between two documents. The RankSVM model can be formulated in the energy-based framework by assuming the following energy function:

$$E_\theta(q_i, d_{ij}, r_{ij}) = -f(q_i, d_{ij}; \theta) \quad (9)$$

with a linear feature model

$$f(q_i, d_{ij}; \theta) = \theta^{\mathrm{T}}\mathbf{x_{i,j}} \quad (10)$$

The loss function L_p for a pair of documents d_{ij} and d_{ik} given query q_i is defined as

$$L_p = \max\left(0, 1 + y_{jk}\big(E_\theta(q_i, d_{ij}, r_{ij}) - E_\theta(q_i, d_{ik}, r_{ik})\big)\right) \quad (11)$$

where y_{jk} is an indicator variable. If document d_{ij} is preferred over d_{ik} (i.e., $r_{ij} > r_{ik}$) given query q_i, $y_{jk} = 1$; otherwise, $y_{jk} = -1$. The combination of the energy and loss with the L_2 regularizer leads to the following loss function for all the pairwise instances:

$$\min_\theta \sum_{i,(j,k)} \max\left(0, 1 - y_{jk}\theta^{\mathrm{T}}(\mathbf{x_{i,j}} - \mathbf{x_{i,k}})\right) + \lambda||\theta||_2^2 \quad (12)$$

This unconstrained optimization problem is equivalent to the following constrained optimization problem [15]:

$$\min_{\theta,\xi} \tfrac{1}{2}||\theta||_2^2 + C\textstyle\sum_{i,(j,k)} \xi_{i,(j,k)} \quad (13)$$

$$s.t. \quad y_{jk}\theta^{\mathrm{T}}(\mathbf{x_{i,j}} - \mathbf{x_{i,k}}) \geq 1 - \xi_{i,(j,k)} \quad (14)$$

$$\xi_{i,(j,k)} \geq 0 \quad (15)$$

where $C = \frac{1}{2\lambda}$. This is the objective function of the RankSVM model [10].

It is worth noting that the loss in Eqn.(11) is different from the hinge loss used in EBMs shown in Table 2. As discussed in the beginning of Section 4, in the energy-based ranking framework, we aim to reshape the energy function over different instances (i.e., documents) while the existing EBMs usually focus on the energy function of a single instance but with different output values.

4.2.2 RankNet

RankNet [3] is one of the learning-to-rank algorithms used by commercial search engines [16]. It is also based on the comparison of a pair of documents. Let us define a loss functional of the energy functions as follows:

$$L_p = \log\left(1 + \exp\left(E_\theta(q_i, d_{ij}, r_{ij}) - E_\theta(q_i, d_{ik}, r_{ik})\right)\right) \quad (16)$$

where d_{ij} is preferred over d_{ik} for query q_i. The following energy function can be used:

$$E_\theta(q_i, d_{ij}, r_{ij}) = -f(q_i, d_{ij}; w) \quad (17)$$

$$= -f(\sum_s w_s f_s(\sum_t w_{st}x_{(t)} + b_s) + b) \quad (18)$$

where f is a three layer neural network with a single output node. $x_{(t)}$ denotes the t-th element of input $\mathbf{x_{i,j}}$, w_{st}, and b_s, and f_s denote the weight, bias, and activation function of the first layer, respectively, w_s, b, and f denote the weight, bias, and activation function of the second layer, respectively. The activation functions are usually sigmoid functions. By plugging the energy function in Eqn.(17) into the loss in Eqn.(16), we obtain the following optimization problem:

$$\min_w \sum_{i,(j,k)} \log\left(1 + \exp\left(f(q_i, d_{ik}; w) - f(q_i, d_{ij}; w)\right)\right) \quad (19)$$

which is equivalent to the objective function in RankNet. In fact, this objective is also equivalent to the Bayesian Personalized Ranking (BPR) optimization criterion [24] (if f is factorized as the product of user and item latent factors), which is widely used in recommender systems for dealing with implicit feedback.

4.3 Listwise

The listwise approach addresses the ranking problem in a more natural way. Specifically, it takes ranked lists as instances in the learning process. The group structure of ranking is maintained. In this section, we study a representative listwise L2R model: ListMLE [29], which exploits the Plackett-Luce (PL) model studied in statistics. PL model defines a probability distribution over permutations of objects, referred to as permutation probability. Let π denote a permutation (ranked list) of the objects and $\pi^{-1}(i)$ denote the object in the i^{th} rank (position) in π. Further suppose that there are non-negative scores assigned to the objects. Let $s = \{s_1, s_2, ..., s_n\}$ denotes the scores of the objects. The PL model defines the probability of permutation π based on scores s as follows.

$$P_s(\pi) = \prod_{i=1}^{n} \frac{s_{\pi^{-1}(i)}}{\sum_{j=i}^{n} s_{\pi^{-1}(j)}} \quad (20)$$

The probabilities of permutations naturally form a probability distribution. In document ranking, given feature vectors $\mathbf{z_1}, \mathbf{z_2}, ..., \mathbf{z_n}$, the top k probability of subgroup $g[\mathbf{z_1}, \mathbf{z_2}, ..., \mathbf{z_n}]$ is calculated as

$$P_s(g[\mathbf{z_1}, \mathbf{z_2}, ..., \mathbf{z_n}]) = \prod_{j=1}^{k} \frac{\exp\left(s(\mathbf{z_j}; \theta)\right)}{\sum_{t=j}^{n} \exp\left(s(\mathbf{z_t}; \theta)\right)} \quad (21)$$

ListMLE maximizes the likelihood of the ground truth ranked lists, which is equivalent to minimizing the following energy-based loss function:

$$L = -\sum_{i=1}^{m} \log \prod_{j=1}^{k} \frac{\exp\big(-E(x_{i,\pi_i^{-1}(j)};\theta)\big)}{\sum_{t=j}^{n_i} \exp\big(-E(x_{j,\pi_i^{-1}(t)};\theta)\big)} \quad (22)$$

where $E(x_{i,\pi_i^{-1}(j)};\theta) = -f(x_{i,\pi_i^{-1}(j)};\theta)$ and f is a neural network model with parameter θ. π_i is the ranking according to the ground truth ranked list for query q_i.

Let us investigate the loss function L_q for query q_i as follow:

$$\begin{aligned} L_q &= -\log \prod_{j=1}^{k} \frac{\exp\big(-E(x_{i,\pi_i^{-1}(j)};\theta)\big)}{\sum_{t=j}^{n_i} \exp\big(-E(x_{i,\pi_i^{-1}(t)};\theta)\big)} \\ &= \sum_{j=1}^{k} E(x_{i,\pi_i^{-1}(j)};\theta) + F(\pi, E;\theta) \end{aligned} \quad (23)$$

where $F(\pi, E;\theta)$ is the contrastive term defined as

$$F(\pi, E;\theta) = \sum_{j=1}^{k} \log \sum_{t=j}^{n_i} \exp\big(-E(x_{i,\pi_i^{-1}(t)};\theta)\big) \quad (24)$$

Based on Eqn.(23), we can explain ListMLE in the energy-based ranking framework as follows. To minimize the loss function in Eqn.(23), we need to "push down" the energies of the top k documents while "pull up" all the energies of the contrastive term (since it is the decreasing function of the energies). For the top i^{th} position, the energies of all the documents below i and including i are pulled up due to the contrastive term, but the energy at the i-th position is pushed down harder by the first term. This can be seen in the expression of the gradient:

$$\begin{aligned} \frac{\partial L_q}{\partial \theta} &= \sum_{j=1}^{k} \frac{\partial E(x_{i,\pi_i^{-1}(j)};\theta)}{\partial \theta} \\ &- \sum_{j=1}^{k} \sum_{t=j}^{n_i} \frac{\partial E(x_{i,\pi_i^{-1}(t)};\theta)}{\partial \theta} P(x_{i,\pi_i^{-1}(t)};\theta) \end{aligned} \quad (25)$$

where

$$P(x_{i,\pi_i^{-1}(t)};\theta) = \frac{\exp\big(-E(x_{i,\pi_i^{-1}(t)};\theta)\big)}{\sum_{t=j}^{n_i} \exp\big(-E(x_{i,\pi_i^{-1}(t)};\theta)\big)} \quad (26)$$

Thus, for each top position i, the contrastive term pulls up on the energy of each document (below or including i) with a force proportional to the negative energy of that document under the model.

5. NEW ENERGY-BASED RANKING MODELS

The energy-based ranking framework establishes the link between learning to rank and EBMs. The existing research in EBMs (e.g., various loss and energy functions) can be readily utilized to solve ranking problems. Furthermore, the energy-based perspective may provide new insights and sensible intuitions to devise novel ranking models. The training of EBMs is essentially the process of reshaping the energy surface. In the pointwise approaches, the energies of correct relevance labels should be decreased, and the energies of incorrect labels should be increased, particularly if they are lower than that of the correct labels. In the pairwise and listwise approaches, we look at the energies of more than a single document. The energies of desired documents are decreased, and the energies of undesired documents are increased. Figure 1 and Figure 2 illustrate the training processes in energy-based ranking. In this section, we present one new model for each of the three representative L2R approaches: pointwise, pairwise, and listwise, respectively.

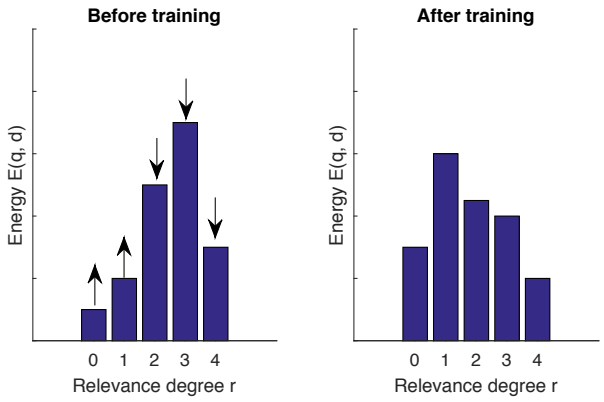

Figure 1: The effect of training on the energy surface in the pointwise case. The energy of the correct relevance label is decreased, and the energies of incorrect labels are increased.

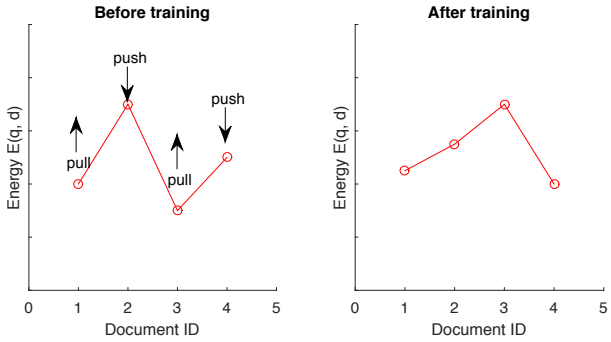

Figure 2: The effect of training on the energy surface in the pairwise and listwise cases. The energies of the desired documents are decreased, and the energies of the undesired documents are increased.

5.1 Pointwise

While many traditional classification and regression models were applied to learning to rank, some energy-based loss functions have not been explored for ranking problems. In this section, we utilize the square-exponential loss (in Table 2) which has demonstrated impressive effectiveness in computer vision applications [13, 5, 20]. The per-query loss functional is defined as follows

$$L_{\text{sq\_exp}} = \sum_{j=1}^{D_i} \Big(E_\theta(q_i, d_{ij}, r_{ij})\Big)^2 + \gamma \exp\Big(-E_\theta(q_i, d_{ij}, \bar{r_{ij}})\Big)$$

where $\bar{r_{ij}}$ is the most offending label for d_{ij} given q_i (i.e., the label that has the lowest energy among all incorrect labels). This loss function aims to push down the energy of correct predictions towards zero while push up the energy of incorrect predictions. To the best of our knowledge, no prior work has applied the square-exponential loss to ranking problems.

The energy function can be defined as the absolute value of the difference between the predicted relevance label and ground truth

label of the document as follows:

$$E_\theta(q_i, d_{ij}, r_{ij}) = ||f(q_i, d_{ij}; \theta) - r_{ij}||_1 \quad (27)$$

where f is a discriminant function as defined in Section 4.1. In the experiments, we assume a simple linear model $f(q_i, d_{ij}; \theta) = \theta^{\mathrm{T}} \mathbf{x_{i,j}}$. We can use the stochastic gradient descent (SGD) algorithm to update the parameters θ as follows

$$\theta := \theta - \eta \frac{\partial E_\theta(q_i, d_{ij}, r_{ij})}{\partial \theta} \Big(E_\theta(q_i, d_{ij}, r_{ij}) - \gamma \exp\big(-E_\theta(q_i, d_{ij}, r_{ij}^-)\big) \Big)$$

where η is a positive learning rate.

5.2 Pairwise

For the pairwise model, we propose to adapt the learning vector quantization (LVQ2) loss functional (in Table 2), which has achieved excellent results in discriminatively training sequence labeling systems, particularly speech recognition systems [7, 18, 11]. The loss functional L_p for a pair of documents d_{ij} and d_{ik} given query q_i is defined as

$$L_{\text{lvq2}} = \min\Big(M, \max\big(0, E_\theta(q_i, d_{ij}, r_{ij}) - E_\theta(q_i, d_{ik}, r_{ik})\big)\Big)$$

where d_{ij} is preferred over d_{ik} for query q_i. Such a loss functional encourages the energy $E_\theta(q_i, d_{ij}, r_{ij})$ of the desired document to be lower than the energy $E_\theta(q_i, d_{ik}, r_{ik})$ of the other document with a margin of zero.

We can use the same energy function with that for RankSVM and RankNet, as defined in Eqn.(9) and Eqn.(17) in Section 4.2. To estimate the parameters θ, we apply the stochastic gradient descent (SGD) update rule as follows given each pair of documents:

$$\theta := \theta - \eta \Big(\frac{\partial E_\theta(q_i, d_{ij}, r_{ij})}{\partial \theta} - \frac{\partial E_\theta(q_i, d_{ik}, r_{ik})}{\partial \theta} \Big) \quad \text{if} \quad 0 \le E_\theta(q_i, d_{ij}, r_{ij}) - E_\theta(q_i, d_{ik}, r_{ik}) \le M \quad (28)$$

where η is the learning rate. Such gradient descent essentially takes steps proportional to the negative of the difference between the gradients of the energies of the document pair.

5.3 Listwise

Section 4.3 illustrates the ListMLE model from the energy-based point of view. It essentially pushes down the energies of the top k documents while pulls up the energies of ALL the documents given a query. Inspired by this observation and Eqn.(25), we devise a new listwise approach by defining the following energy function for a ranked list π_i:

$$E_{list}(\pi_i; \theta) = \sum_{j=1}^{k} E(x_{i,\pi_i^{-1}(j)}; \theta) \quad (29)$$

$$- \sum_{j=1}^{k} \sum_{t=j}^{n_i} E(x_{i,\pi_i^{-1}(t)}; \theta) P(x_{i,\pi_i^{-1}(t)}; \theta) \quad (30)$$

where $E(x_{i,\pi_i^{-1}(j)}; \theta)$ is the energy of the individual document $x_{i,\pi_i^{-1}(j)}$ as defined in Section 4.3 for ListMLE. The listwise energy function $E_{list}(\pi_i; \theta)$ has two parts. The first part (i.e., Eqn.(29)) is the sum of the energies of the top k documents. The second part (i.e., Eqn.(30), the contrastive term) is the sum of the expected energies of the k ranked sublists. These sublists include the documents from the j^{th} position to the end, where $j \in [1, k]$, respectively. In other words, $E_{list}(\pi_i; \theta)$ is the sum of top-k discrepancy between the energy of the document in the top j^{th} position and the expectation of energy of documents that are ranked from the j^{th} position to the bottom. If we want to minimize the energy $E_{list}(\pi_i; \theta)$ of the whole list, we need to lower the energy of the first part and raise the energy of the second part. As a result, this will make the top k documents more distinguishable from the rest of the documents in the ranked list.

Different from ListMLE, we define $P(x_{i,\pi_i^{-1}(t)}; \theta)$ only based on the rank position t:

$$P(x_{i,\pi_i^{-1}(t)}; \theta) = \begin{cases} \frac{1}{1+\sum_{t=2}^{n_i} 1/\log(t)}, & t = 1 \\ \frac{1/\log(t)}{1+\sum_{t=2}^{n_i} 1/\log(t)}, & t > 1 \end{cases} \quad (31)$$

This is motivated by the discounting factor in Normalized Discounted Cumulative Gain (NDCG). The rank position based probability does not depend on the features of individual documents or parameters and thus it is more efficient than that defined in Eqn.(26)) for ListMLE.

Given the energy function $E_{list}(\pi_i; \theta)$ defined over the list π_i, we can use various loss functionals of the energy-based models, e.g., the LVQ2 loss as follows:

$$\mathcal{L}_{\text{lvq2-list}} = \frac{1}{m} \sum_{i=1}^{m} \min\Big(M, \max\big(0, E_{list}(\pi_i; \theta)\big)\Big) \quad (32)$$

where m is the total number of queries/ranked lists. This loss aims to minimize $E_{list}(\pi_i; \theta)$ by some margin M. We can use the stochastic gradient descent to update the parameters as follows:

$$\theta := \theta - \eta \sum_{j=1}^{k} \Big(\frac{\partial E(x_{i,\pi_i^{-1}(j)}; \theta)}{\partial \theta} - \sum_{t=j}^{n_i} \frac{\partial E(x_{i,\pi_i^{-1}(t)}; \theta)}{\partial \theta} P(x_{i,\pi_i^{-1}(t)}; \theta) \Big) \quad \text{if} \quad 0 \le E_{list}(\pi_i; \theta) \le M$$

It is worth noting that the above model is just one example of listwise approaches based on the energy oriented perspective of ListMLE. In the future work, we will explore to cast other existing listwise L2R models into the energy-based ranking framework, which may offer further insights into devising new listwise techniques.

6. EXPERIMENTS

6.1 Testbeds

We use the benchmark datasets from the LETOR 4.0 learning to rank testbeds[1]. The datasets includes two tasks, MQ2007 and MQ2008, which are drawn from the data in TREC 2007 and 2008 collection. Table 3 provides the statistics about the two corpora. The size of the MQ2007 dataset is larger than that of MQ2008 in terms of the number of queries (1,692 vs. 784) and query-document pairs (69,623 vs. 15,211). The average number of documents per query in the MQ2007 dataset is two times larger than that in the MQ2008 dataset, while the number of documents per query in the MQ2008 dataset is more varied (22.074 vs. 6.684). Each example in the dataset stands for a query-document pair, which is represented by 46 features related to information retrieval such as TF-IDF similarity measures between query and document, PageRank,

[1]http://research.microsoft.com/en-us/um/beijing/projects/letor/letor4dataset.aspx

Table 3: Statistics of the LETOR 4.0 datasets

	MQ2007	MQ2008
#queries	1,692	784
#query-document pairs	69,623	15,211
#Min. documents per query	6	5
#Max. documents per query	147	121
#Avg. documents per query	41.148	19.402
#Std. documents per query	6.684	22.074

and BM25 [22]. The relevance between a query and document is judged on three levels $\{0, 1, 2\}$ with 2 being most relevant.

Each task is partitioned for five-fold cross validation, including training (60%), validation (20%), and test (20%) data sets. All the proposed energy-based approaches as well as the baselines are trained using the entire training data, and their performance is evaluated on the test data. The parameters are determined on the validation data. We use coarse grid search to tune model parameters (see Section 7.1) and report the mean test values and perform statistical significance tests across 30 runs of 5-fold cross validation (see Section 7.3).

6.2 Baselines

We compare the three proposed energy-based learning to rank models against the following state-of-the-art pointwise, pairwise, and listwise learning to rank methods [16]. For pointwise approaches, we choose L_2-regularized linear regression, L_2-regularized logistic regression, and support vector machine (SVM). We rely on the Scikit-learn package [21] to train and test those models with multiple regularization parameters. For pairwise approaches, we use RankNet [3] and RankSVM [10]. RankLib[2] is used to train and test RankNet models with default parameters; and we use SVM^{rank3} to train and test RankSVM[4]. For listwise approaches, we use ListMLE [29] with linear neural network as ranking function. All these baselines are also formulated as energy-based ranking models in Section 4. We will make our source code publicly available.

6.3 Evaluation Metrics

In the experiments, we use the following metrics for evaluation: (1) Precision at position k (P@k) where k is set to 5, 10, 15, and 20, respectively; (2) Mean Average Precision (MAP), which measures the averaged P@k of all queries; (3) Normalized Discount Cumulative Gain at position k (NDCG@k), which measures the ranking quality for each query at position k. We choose k as 5, 10, 15, and 20; (4) Mean Reciprocal Rank (MRR), which measures the averaged rank position of the first relevant document for each query; and (5) Mean Squared Error (MSE), which measures the differences between predicted relevance labels and ground truth labels. This metric is only applicable to evaluate pointwise approaches. We rely on the TREC evaluation script[5] to calculate these metrics.

7. RESULTS

7.1 Parameter Analysis

We first examine the impact of parameters γ, M, and k on our proposed energy-based pointwise, pairwise, and listwise learning to rank approaches introduced in section 5, respectively and thus determine the parameter settings. Figure 3 to 8 show the average Precision and NDCG values at different levels achieved by different parameters across 5 runs of 5 fold cross validation on the MQ2007 and MQ2008 tasks.

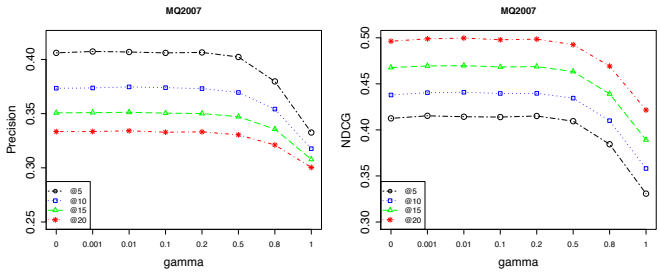

Figure 3: Average metrics in 5 runs of energy-based pointwise approach with different γ on MQ2007 dataset

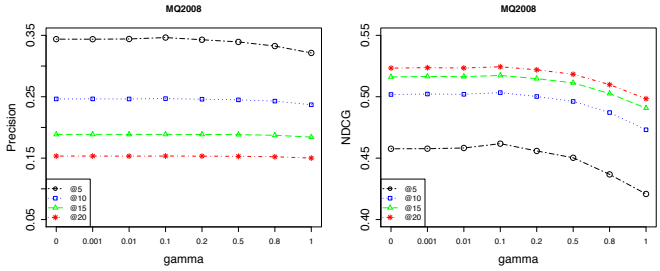

Figure 4: Average metrics in 5 runs of energy-based pointwise approach with different γ on MQ2008 dataset

For pointwise approaches, Figure 3 and 4 show that when γ is set between 0 and 0.2 on the MQ2007 dataset and 0 and 0.1 on the MQ2008 dataset, the precision and NDCG values remain stable, indicating that the exponential component in the square-exponential loss function plays a less important role than the square component does. Those γ values are good choices. When γ becomes larger, the metrics start to decrease quickly except the precision values on the MQ2008 dataset. We select $\gamma = 0.001$ for the MQ2007 dataset and $\gamma = 0.1$ for the MQ2008 dataset based on the performance on the validation set. For pairwise approaches (see Figure 5 and Figure 6), as M increases from 0.001 to 10.0, the precision and NDCG scores at all levels decrease on both datasets. Thus, we set $M = 0.001$. For listwise approaches (see Figure 7 and 8), we have two parameters for tuning, M and k. For MQ2007 dataset, the precision and NDCG values are increasing when k increases from 1 to 5, and they start to decrease when $k = 10$. Therefore, we set $k = 5$. While M is growing until $M = 1.0$, both precision and

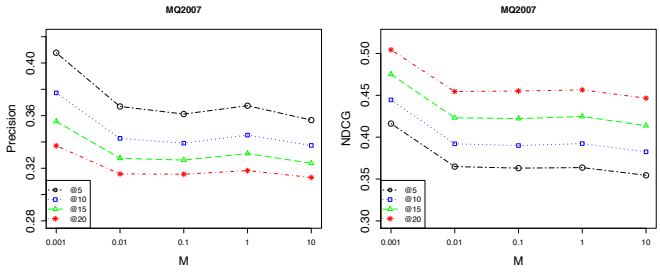

Figure 5: Average metrics in 5 runs of energy-based pairwise approach with different M on MQ2007 dataset

[2]https://sourceforge.net/p/lemur/wiki/RankLib/
[3]https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html
[4]http://research.microsoft.com/en-us/um/beijing/projects/letor/LETOR4.0/Baselines/RankSVM-Struct.html
[5]http://trec.nist.gov/trec_eval/

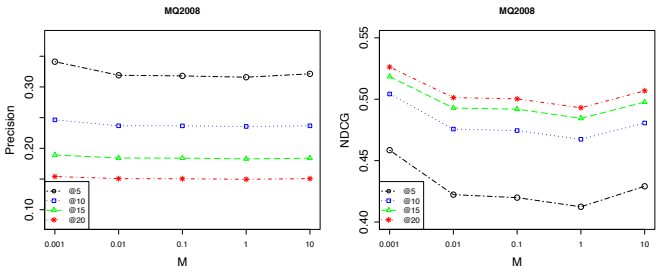

Figure 6: Average metrics in 5 runs of energy-based pairwise approach with different M on MQ2008 dataset

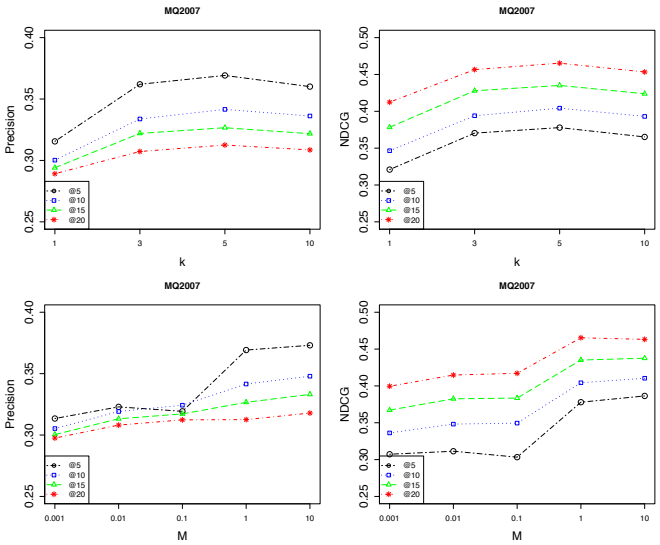

Figure 7: Average metrics in 5 runs of energy-based listwise approach with different M and k on MQ2007 dataset

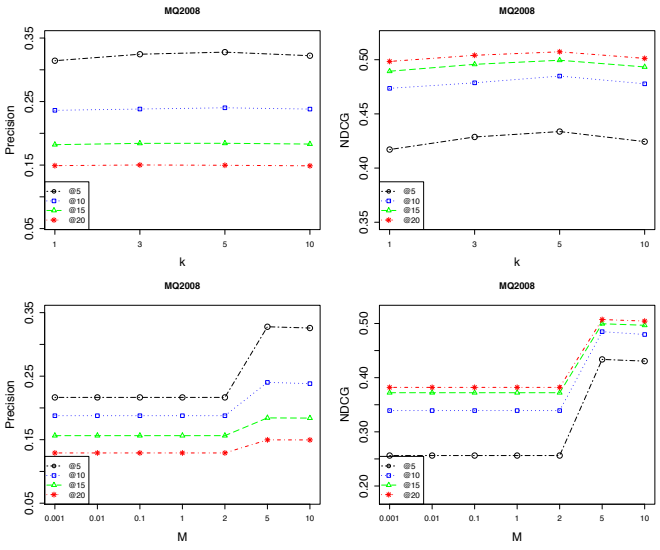

Figure 8: Average metrics in 5 runs of energy-based listwise approach with different M and k on MQ2008 dataset

Table 4: Parameters for Energy-based Approaches

Model	Parameter	MQ2007	MQ2008
Pointwise	# of iterations	20	20
	L_2-Regularizer	0.1	0.1
	Learning rate	5e-4	5e-4
	γ	0.001	0.001
Pairwise	# of iterations	10	10
	L_2-Regularizer	0.1	0.1
	Learning rate	1e-4	1e-4
	M	1.0	0.1
	k	5	1
Listwise	# of iterations	10	10
	L_2-Regularizer	0.1	0.1
	Learning rate	1e-4	1e-5
	M	1.0	5.0
	k	5	5

NDCG scores are increasing and they slighly decrease when M is getting larger. Thus, we choose $M = 1.0$. For MQ2008 dataset, the NDCG values decrease and precision values remain relatively stable when k is growing, but they reach the peak when k is set to 5, so we set $k = 5$. Regarding M, both precision and NDCG values remain unchanged when M is increasing from 0.001 to 2.0. The metrics start to increase when M is set to 5.0 and decrease slightly afterwards. Thus, $M = 5.0$ seems a good choice.

We also experiment with different settings of other parameters, e.g., L_2-regularizer, learning rate, and number of SGD iterations, but leave out their analysis. The parameter settings obtained for our proposed methods are shown in Table 4.

7.2 Analysis of Shaping Energy Surface

Recall that training an energy-based learning to rank model aims to shape an energy function that produces the best ranking results given a set of documents retrieved by a query. The best ranking list is expected to have the lowest energy than all other permutations. We demonstrate the changes of energy function that is defined on the listwise approach introduced in Section 5.3. Figure 9 plots the curve of decreasing rate of energy of sum of documents ranked in top-5 positions (Eqn.(29)) and that of contrastive documents that are ranked from the 5-th position to the bottom (Eqn.(30)) over 30 rounds of SGD on the MQ2008 dataset. As we can see, at the beginning of the SGD algorithm, the energy of desired documents are decreasing sharply (over 100%), and the energy of undesired documents are also decreasing but with a very tiny rate (around 1%), which is hardly seen in the figure. After 10 rounds of SGD, the decreasing rate of the energy of desired documents reduces to 10%, while the decreasing rate of energy of undesired documents remains consistently low. The SGD algorithm ends up with the energy of desired documents decreased to just 3% of the energy obtained at the beginning, and energy of undesired documents decreased to only 70% compared with the original energy. Hence, the energy function defined over all ranked lists are decreasing during the training phase, due to that the energy of all documents are pushed down, but not as hard as it is pushed down by the first term.

7.3 Baseline Comparison

The results for the two tasks achieved by different learning to rank methods obtained using the parameter settings in Section 7.1 are shown in Table 5 and Table 6. Boldface stands for best performance with respect to each evaluation metric for pointwise, pairwise, and listwise learning to rank methods, respectively. As pair-

Table 5: Average performance on MQ2007 dataset

Method	MAP	MRR	P@5	P@10	P@15	P@20	N@5	N@10	N@15	N@20	MSE
Linear Regression	0.4257	0.5393	0.3797	0.3546	0.3378	0.3238	0.3803	0.4087	0.4399	0.4712	0.3094
Logistic Regression	0.3393	0.4388	0.2915	0.2825	0.2749	0.2684	0.2746	0.2985	0.3249	0.3540	0.4196
SVM	0.3308	0.4129	0.2805	0.2764	0.2708	0.2659	0.2570	0.2835	0.3113	0.3414	0.4241
Energy-based Pointwise	**0.4520**	**0.5622**	**0.4074**	**0.3737**	**0.3509**	**0.3335**	**0.4153**	**0.4402**	**0.4695**	**0.4989**	**0.3060**
RankNet	0.4279	0.5316	0.3727	0.3480	0.3337	0.3206	0.3755	0.4042	0.4372	0.4695	-
RankSVM	**0.4637**	**0.5762**	**0.4133**	**0.3810**	**0.3587**	**0.3380**	**0.4246**	**0.4517**	**0.4825**	**0.5091**	-
Energy-based Pairwise	0.4583	0.5672	0.4078	0.3773	0.3556	0.3371	0.4162	0.4445	0.475	0.5043	-
ListMLE	0.3967	0.5048	0.3412	0.3218	0.3116	0.3023	0.3457	0.3748	0.4074	0.4395	-
Energy-based Listwise	**0.4212**	**0.5306**	**0.3692**	**0.3416**	**0.3266**	**0.3125**	**0.3779**	**0.4043**	**0.4351**	**0.4653**	-

Table 6: Average performance on MQ2008 dataset

Method	MAP	MRR	P@5	P@10	P@15	P@20	N@5	N@10	N@15	N@20	MSE
Linear Regression	0.4332	0.4958	0.3214	0.2367	0.1845	0.1508	0.4231	0.4733	0.4917	0.5002	0.2878
Logistic Regression	0.3290	0.4023	0.2306	0.1916	0.1584	0.1299	0.2919	0.3666	0.3975	0.4069	0.3725
SVM	0.3124	0.3670	0.2194	0.1885	0.1569	0.1293	0.2645	0.3463	0.3787	0.3885	0.3769
Energy-based Pointwise	**0.4651**	**0.5250**	**0.3464**	**0.2469**	**0.1887**	**0.1534**	**0.4618**	**0.5034**	**0.5173**	**0.5244**	**0.2764**
RankNet	0.4360	0.4893	0.3168	0.2329	0.1816	0.1481	0.4181	0.4694	0.4875	0.4960	-
RankSVM	**0.4707**	0.5266	**0.3477**	**0.2487**	**0.1894**	0.1536	**0.4630**	**0.5080**	**0.5214**	**0.5286**	-
Energy-based Pairwise	0.4677	**0.5279**	0.3412	0.2464	0.1892	**0.1541**	0.4585	0.5043	0.5184	0.5262	-
ListMLE	0.4308	0.4877	0.3128	0.2353	0.1821	0.1491	0.4159	0.4723	0.4889	0.4980	-
Energy-based Listwise	**0.4445**	**0.5046**	**0.3278**	**0.2402**	**0.1843**	**0.1497**	**0.4337**	**0.4849**	**0.4995**	**0.5073**	-

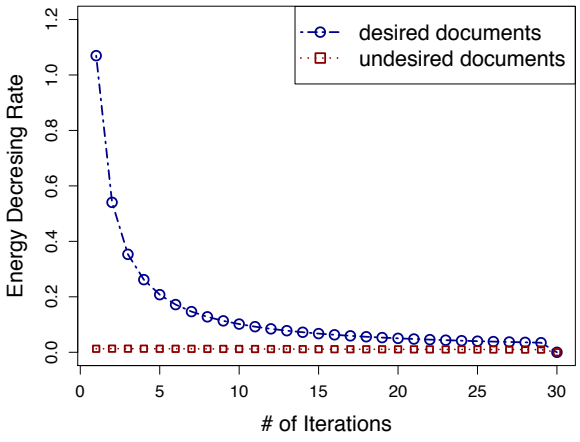

Figure 9: The effect of training on the energy surface in the listwise case on MQ2008 dataset

wise and listwise L2R approaches are not designed to predict relevance for documents, we do not report the MSEs obtained in these two cases.

In comparison with the results achieved on the two tasks, the precision scores (MAP and P@N) obtained on the MQ2007 dataset are consistently higher than that obtained on the MQ2008 dataset for all L2R approaches. This may be due to the fact that the number of query-document pairs of the MQ2007 dataset is four times larger than that of the MQ2008 dataset. Therefore, more training instances contribute to the better performance of identifying relevant documents. On the contrary, the scores (MRR and NDCG@N) achieved on the MQ2007 dataset is lower than that achieved on the MQ2008 dataset, which may be caused by the larger number of averaged documents per query in the MQ2007 dataset (see Table 3).

For pointwise approaches, linear regression yields better results in terms of all the evaluation metrics on both tasks than the other two pointwise baselines, logistic regression and SVM do. Our proposed energy-based pointwise method achieves the best performance over all the pointwise baselines with statistically significant improvement ($p < 0.001$) for all precision and ranking evaluation metrics. On the MQ2007 dataset, the energy based pointwise approach achieves more than 20% of precision values than logistic regression and SVM do, and 3% than linear regression does. In terms of NDCG values, the proposed model can obtain more than 40% of improvement compared with logistic regression and SVM, and 5% of improvement over linear regression. On the MQ2008 dataset, the improvement on the precision values are over 15% compared with logistic regression and SVM, and over 1% compared with linear regression. The model also yields significant MSEs on both datasets with $p = 0.05$ and $p < 0.001$. It outperforms linear regression by 1.1% on the MQ2007 dataset and 3.96% on the MQ2008 dataset.

For pairwise approaches, RankSVM shows better results than RankNet does for all evaluation metrics on both tasks. Our proposed energy-based pairwise method outperforms RankNet significantly ($p < 0.001$) on all evaluation metrics on both datasets. The model outperforms RankNet in terms of precision values by at least 5% and NDCG values by 7% on the MQ2007 dataset; and it achieves more than 4% of precision values and 6% of NDCG values on the MQ2008 dataset. The energy-based pairwise approach yields competitive results with RankSVM does with slightly lower precision and ranking scores on both datasets (except MRR and P@20 obtained on the MQ2008 dataset) without significant difference. The difference of all evaluation metrics achieved by RankSVM and our proposed model is just around 1% for both tasks. The best performance obtained by pairwise approach is higher or competitive with the best performance obtained by pointwise L2R approaches.

For listwise approaches, the results obtained by ListMLE and our proposed models are slightly inferior to that obtained by pointwise and pairwise approaches for both tasks. Our proposed energy-based listwise approach yields significant results over ListMLE ($p = 0.01$) on both datasets in terms of all precision and NDCG scores. On the MQ2007 dataset, the proposed model outperforms ListMLE by 3% for precision values and 5% for NDCG values. On the MQ2008 dataset, the improvement on the precision values is over 1% except P@20 and that on the NDCG values is over 1%.

In summary, our proposed energy-based pointwise and listwise methods surpass the corresponding learning to rank baselines with significant improvement. For the pairwise approach, the proposed model outperforms RankNet with a significant margin and achieves competitive results with RankSVM. Our results differ from the previous ones on LETOR in the following way. The energy-based pairwise method achieves the best performance consistently across the two tasks, despite the fact that the previous studies favored listwise approaches [29, 4]. This may be due to the fact that the tasks used for experiments are different. The dataset used for their study is LETOR 3.0, while we use LETOR 4.0 for experiments. Another possible reason is that we only use linear models to construct our energy functions in the experiments. We leave experimenting with nonlinear models and larger testbeds in our future work.

8. CONCLUSION AND FUTURE WORK

In this paper, we establish the link between learning to rank and energy-based learning. We cast various existing L2R models in a unified energy-based ranking framework. Moreover, we present several new energy-based ranking models based on the established link. The experiments are conducted on the LETOR 4.0 benchmark datasets and demonstrate the effectiveness of the proposed models.

This work is an initial step towards a promising research direction. In the future work, we will cast more sophisticated L2R models in the energy-based framework and propose new ranking algorithms accordingly. Furthermore, we plan to investigate latent variable architectures where energy functions depend on a set of hidden variables whose correct values are unobserved. The use of latent variables in ranking may be able to model the hidden characteristics of queries, documents, and relevance. We will also explore deep architectures based on the energy-based models with hierarchical latent variables. Last but not the least, the existing energy-based learning theories may not be directly applied to ranking. We will conduct theoretical analysis for energy-based ranking such as necessary or sufficient conditions for energy and loss functions and generalization ability and statistical consistency of energy-based ranking models. The new theories may utilize the energy-based perspective to help explain and justify the choices of loss functions in L2R.

9. REFERENCES

[1] A. Bordes, X. Glorot, J. Weston, and Y. Bengio. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 94(2):233–259, 2014.

[2] P. Brakel, D. Stroobandt, and B. Schrauwen. Training energy-based models for time-series imputation. *JMLR*, 14(1):2771–2797, 2013.

[3] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML*, pages 89–96. ACM, 2005.

[4] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *ICML*, pages 129–136. ACM, 2007.

[5] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, volume 1, pages 539–546. IEEE, 2005.

[6] D. Cossock and T. Zhang. Subset ranking using regression. In *Learning theory*, pages 605–619. Springer, 2006.

[7] X. Driancourt, L. Bottou, and P. Gallinari. Learning vector quantization, multi layer perceptron and dynamic programming: comparison and cooperation. In *IJCNN*, volume 2, pages 815–819. IEEE, 1991.

[8] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *JMLR*, 4:933–969, 2003.

[9] N. Heess, D. Silver, and Y. W. Teh. Actor-critic reinforcement learning with energy-based policies. In *EWRL*, pages 43–58, 2012.

[10] T. Joachims. Optimizing search engines using clickthrough data. In *SIGKDD*, pages 133–142, 2002.

[11] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang. A tutorial on energy-based learning. *Predicting structured data*, 1:0, 2006.

[12] Y. LeCun, S. Chopra, M. Ranzato, and F. J. Huang. Energy-based models in document recognition and computer vision. In *ICDAR*, volume 7, pages 337–341, 2007.

[13] Y. LeCun and F. J. Huang. Loss functions for discriminative training of energy-based models. In *AISTATS*, 2005.

[14] J. Lei, G. Li, D. Tu, and Q. Guo. Convolutional restricted boltzmann machines learning for robust visual tracking. *Neural Computing and Applications*, 25(6):1383–1391, 2014.

[15] H. Li. Learning to rank for information retrieval and natural language processing. *Synthesis Lectures on Human Language Technologies*, 7(3):1–121, 2014.

[16] T.-Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[17] T.-Y. Liu, J. Xu, T. Qin, W. Xiong, and H. Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *SIGIR Workshop on Learning to Rank for Information Retrieval*, pages 3–10, 2007.

[18] E. McDermott. *Discriminative training for speech recognition*. PhD thesis, Waseda University, 1997.

[19] R. Memisevic and G. E. Hinton. Learning to represent spatial transformations with factored higher-order boltzmann machines. *Neural Computation*, 22(6):1473–1492, 2010.

[20] M. Osadchy, Y. L. Cun, and M. L. Miller. Synergistic face detection and pose estimation with energy-based models. *JMLR*, 8:1197–1215, 2007.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *JMLR*, 12:2825–2830, 2011.

[22] T. Qin and T.-Y. Liu. Introducing letor 4.0 datasets. *arXiv preprint arXiv:1306.2597*, 2013.

[23] M. Ranzato, Y.-L. Boureau, S. Chopra, and L. Yann. A unified energy-based framework for unsupervised learning. In *AISTATS*, 2007.

[24] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461. AUAI Press, 2009.

[25] A. R. Sankar and V. N. Balasubramanian. Similarity-based contrastive divergence methods for energy-based deep learning models. In *ACML*, pages 391–406, 2015.

[26] G. W. Taylor and G. E. Hinton. Factored conditional restricted boltzmann machines for modeling motion style. In *ICML*, pages 1025–1032. ACM, 2009.

[27] M. N. Volkovs and R. S. Zemel. Boltzrank: learning to maximize expected ranking gain. In *ICML*, pages 1089–1096. ACM, 2009.

[28] M. Welling, M. Rosen-Zvi, and G. E. Hinton. Exponential family harmoniums with an application to information retrieval. In *NIPS*, pages 1481–1488, 2004.

[29] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach to learning to rank: theory and algorithm. In *ICML*, pages 1192–1199. ACM, 2008.

[30] J. Xu and H. Li. Adarank: a boosting algorithm for information retrieval. In *SIGIR*, pages 391–398. ACM, 2007.

[31] X. Zhang. *PAC-Learning for Energy-based Models*. PhD thesis, Courant Institute of Mathematical Sciences New York, 2013.