Unsupervised Ranking faces one critical challenge in evaluation applications, that is, no ground truth is available. When PageRank and its variants show a good solution in related objects, they are applicable only for ranking from link-structure data. In this work, we focus on unsupervised ranking from multi-attribute data which is also common in evaluation tasks. To overcome the challenge, we propose five essential meta-rules for the design and assessment of unsupervised ranking approaches: scale and translation invariance, strict monotonicity, compatibility of linearity and nonlinearity, smoothness, and explicitness of parameter size. These meta-rules are regarded as high level knowledge for unsupervised ranking tasks. Inspired by the works in [12] and [35], we propose a ranking principal curve (RPC) model, which learns a one-dimensional manifold function to perform unsupervised ranking tasks on multi-attribute observations. Furthermore, the RPC is modeled to be a cubic Bézier curve with control points restricted in the interior of a hypercube, complying with all the five meta-rules to infer a reasonable ranking list. With control points as model parameters, one is able to understand the learned manifold and to interpret and visualize the ranking results. Numerical experiments of the presented RPC model are conducted on two open datasets of different ranking applications. In comparison with the state-of-the-art approaches, the new model is able to show more reasonable ranking lists.

Index Terms—Unsupervised ranking, multi-attribute, meta-rules, data skeleton, principal curves, Bézier curves

1 INTRODUCTION

Ranking is a fundamental problem in many applications, such as information retrieval and social choice. From the viewpoint of machine learning, ranking can be performed in an either supervised or unsupervised way as shown in Table 1. When supervised ranking [3] is able to evaluate the ranking performance from the given ground truth, unsupervised ranking seems more challenging because no ground truth label is available. Modelers or users will encounter a more difficult issue below:

“How can we insure that the ranking list from the unsupervised ranking is reasonable or proper?”

Although many unsupervised ranking strategies appeared recently, they took supervised evaluation measures like normalized discounted cumulative gain (NDCG) [4], mean average precision (MAP) [18], and recall precision of top-k lists. For example, unsupervised rank aggregation methods for meta-search are evaluated on LETOR [13] with NDCG and MAP. However, in cases of other ranking types that is there is no ground-truth label, like countries and journals ranking, unsupervised ranking are still stuck in problems of evaluation.

PageRank [9] is one of accepted unsupervised ranking methods for web search since it makes full use of domain knowledge of web links, following the principle of “let the data speak for themselves” [33]. However, unsupervised ranking from multi-attribute objects does not have the same domain knowledge of web links such that PageRank is not suitable any more. Yet it highlights a possible solution. Unsupervised ranking from multi-attribute objects also has domain knowledge which can be made use of. In this work, we formally formulate this domain knowledge, which has been widely used in ranking problems (both supervised and unsupervised), into five essential meta-rules (or preferred features). These meta-rules can act not only as a high level evaluation of unsupervised ranking lists, but also as a guidance of designing ranking functions (Fig. 1).

Unsupervised ranking strategies for multi-attribute objects can be broadly divided into two categories: rank aggregation and structure based. Borda’s methods [19] and multinomial preference model (MPM) [21] fall into the first category which aggregate several ranking lists. Although Borda’s methods are simple to carry out, they are regardless of the specific distributions of different datasets. MPM pays attention to the score distribution, but it is an implicit rank aggregation method such that it hardly tells whether it is order-preserving (see Definition 2) which is an important requirement for ranking [11], [34]. The first principal component analysis (PCA) [17] and elastic map (Elmap) [28], fall into another category which try to depict the data skeleton as “ranking coordinate”. However, PCA is suitable for linearly-shaped datasets but not nonlinearly-shaped ones (e.g., Fig. 5 in Section 5) while Elmap has no guarantee of order-preserving. Yet, replacing Elmap directly by the other
Suppose we want to evaluate life qualities of countries with a principal curve based on two attributes: Life Expectancy at Birth (LEB) and GDP.

Each country is a data point in the two-dimensional plane of LEB and GDP. If the principal curve is approximated by a polyline as in Fig. 2a, the piece of the horizontal line is not strictly monotone. It makes the same ranking solution for \[ x_1 = (58, 1.4) \] and \[ x_2 = (58, 16.2) \] but \[ x_2 \] should be ranked higher than \[ x_1 \].

For a general principal curve like the curve in Fig. 2b which is not monotone, two pairs of points are ordered unreasonably. The pair, \( x_1 = (74, 40.2) \) and \( x_2 = (82, 40.2) \), are put in the same place of the ranking list since they are projected to the same point which has the vertical tangent line to the curve. But \( x_1 \) should be ranked higher for its higher LEB than \( x_2 \).

Another pair, \( x_3 = (75, 62.5) \) and \( x_4 = (81, 64.8) \), are also put in the same place but apparently \( x_3 \) should be ranked higher than \( x_4 \).

The following points highlight the main contributions of this paper:

- We propose five meta-rules for unsupervised ranking, which serve as high-level guidance in the design and assessment of unsupervised ranking approaches for multi-attribute objects. We justify that the five meta-rules are essential in model design and model selection for unsupervised ranking, but unfortunately some of them were overlooked by most unsupervised ranking approaches.

- A ranking principal curve (RPC) model is presented for unsupervised ranking from multi-attribute numerical observations of objects. The presented model, parameterized by a cubic Bézier curve, can satisfy all of five meta-rules for ranking tasks, while most existing approaches fail to follow them. What’s more, the parameterization makes it more easy to interpret and visualize of ranking results.

- We develop the RPC learning algorithm, and theoretically prove the existence of a RPC and convergence of learning algorithm for given multi-attribute objects for ranking. With RPC learning algorithm, reasonable ranking lists for openly accessible data illustrate the good performance of the proposed unsupervised ranking approaches.

### 1.1 Related Works

For the evaluation of supervised ranking performance, NDCG [4] and MAP [18] are two widely used indicators in web search which involve the target ranking labels. But for the evaluation of unsupervised ranking performance where no target labels are available, there is not yet an acceptable evaluation method. Among existing unsupervised ranking methods [19], [21], [23], most of them focus on the search ranking and are evaluated on two datasets of query searching results, TREC and LETOR, with NDCG and MAP. It is domain specific. For other types of unsupervised ranking problems, like countries and journals ranking, how to produce an acceptable ranking result remains a challenge.

The most famous unsupervised ranking is PageRank algorithm which makes full use of backlinks to rate a website for searching [9]. After that, numerous ranking algorithms are developed for web search including heuristic rank aggregation methods [22]. Most of these methods are unsupervised. Ranking on manifolds has provided a new ranking framework [12], [14], [15], [16], which is different from general ranking functions such as ranking aggregation [23]. As one-dimensional manifolds, principal curves are able to perform unsupervised ranking tasks from multi-attribute numerical observations of objects [12]. But not all principal curve models can serve as ranking functions. For example, Elmap can well portray the contour of a molecular surface [28] but would bring about a biased ranking list due to no guarantee of order-preserving [12].

Domain knowledge has been widely used in machine learning fields. For example, ranking knowledge of monotonicity has been used to improve ordinal classification and regression [10], [11], [34]. For unsupervised learning, learning models with constraints of domain knowledge are expected to produce closer learning results to ground truths.
than those without knowledge constraints. An example is unsupervised image segmentation which has used the domain knowledge in segment assessing [24]. For unsupervised ranking tasks, their domain knowledge (like monotonicity) can also help to evaluate and design ranking functions which is the motivation of this work.

1.2 Paper Organization
The rest of this paper is organized as follows. Preliminaries and backgrounds are formalized in Section 2 and Section 3 respectively. In Section 4, five meta-rules are elaborated for unsupervised ranking. In Section 5, the RPC model is defined and formulated with a cubic Bézier curve following all the five meta-rules. RPC learning algorithm is designed in Section 6. To illustrate the effective performance of RPC, applications on real world datasets are carried out in Section 7, prior to summary of this paper in Section 8.

2 PRELIMINARIES
2.1 Principal Curves
The principal curve is the nonlinear extension of the first PCA in the sense of depicting a data skeleton. Let a dataset $X = (x_1, x_2, \ldots, x_n)$, $x_i \in \mathbb{R}^d$. The first PCA summarizes the data with a straight line, $x = \mu + sw + \epsilon$ where $\mu$ is the mean of $X$, $w$ is the line direction, $s$ is the first component score and $\epsilon$ is the residual which is taken as noise in practice. Instead of a line, a principal curve summarizes the data with a smooth curve

$$x = f(s) + \epsilon$$

where $f(s) = (f_1(s), f_2(s), \ldots, f_d(s)) \in \mathbb{R}^d$ and $s \in \mathbb{R}$. The principal curve $f$ was originally defined by Hastie and Stuetzle [26] as a smooth (C$^\infty$) unit-speed ($\|f''\|^2 = 1$) one-dimensional manifold in $\mathbb{R}^d$ satisfying the self-consistency condition

$$f(s) = E(x|s_f(x) = s),$$

where $s = s_f(x) \in \mathbb{R}$ is the largest value so that $f(s)$ has the minimum distance from $x$. Mathematically, $s_f(x)$ is formulated as [26]

$$s_f(x) = \sup \left\{ s : \|x - f(s)\| = \inf_s \|x - f(s)\| \right\}.$$  

A curve $f: \mathbb{R} \rightarrow \mathbb{R}^d$ is called a principal curve if it minimizes the expected squared distance between $x$ and $f$ which is denoted by [27]

$$J(f) = E(\inf_s \|x - f(s)\|^2) = E\|x - f(s_f(x))\|^2.$$  

Following Hastie and Stuetzle [26], researchers afterwards have proposed a variety of principal curve models to perform different tasks [27], [28], [29]. Some of them tried to first approximate the principal curve with a polyline [27] and then smooth it to meet the requirement of smoothness [26]. Some of them tried to find some key points and connect them smoothly to form a principal curve [31]. Therefore, expressions of principal curves are not explicit and turn to be a “black-box” which is hard to interpret. Some definitions of principal curves [30], [32] employed Gaussian mixture model to parametrically formulate the principal curve resulting in model bias. When the principal curve is used to perform a ranking task [28], it is required to be a “white-box” for easy interpretability and to meet the basic requirements of ranking, such as order-preserving.

2.2 Bézier Curves
A Bézier curve is a parametric model which is frequently used in computer graphics to model a smooth curve. The explicit formulation of the Bézier curve of degree $k$ is given [45] parametrically by

$$f(s) = \sum_{r=0}^{k} B_r^k(s)p_r, \ s \in [0, 1]$$

in terms of Bernstein polynomials

$$B_r^k(s) = \binom{k}{r}(1-s)^{k-r} s^r,$$

$$\binom{k}{r} = \frac{k!}{r!(k-r)!}.$$  

In Eq. (4), $p_r \in \mathbb{R}^d$ are control points of the Bézier curve. Taking $p_0$ and $p_k$ to be end points, the Bézier curve is regarded as the nonlinear interpolation of end points.

The Bézier curve has some attractive geometric properties in practice [45], four of which are listed as follows:

- Affine Compatibility. The Bézier construction is compatible with affine transformations. Let $\mathbf{L}$ be an affine map. $\mathbf{L}$ might perform scaling, translation and rotation. The affine transformation on the Bézier curve is directly imposed on the control points

$$\mathbf{L}(f(s)) = \sum_{r=0}^{k} B_r^k(s)\mathbf{L}(p_r).$$

due to $\sum_{r=0}^{k} B_r^k(s) = 1$.

- Invariance under Interval Affine Transformation. The Bézier curve is invariant to interval affine from $[0, 1]$ to $[a, b]$, which reads algebraically

$$f(s) = f\left(\frac{u - a}{b - a}\right) = \sum_{r=0}^{k} B_r^k\left(\frac{u - a}{b - a}\right)p_r, u \in [a, b].$$

- Convex Hull. For $0 \leq s \leq 1$, the Bézier curve lies entirely in the convex hull of its control points.

- Reduced-Order Derivatives. The Bézier curve is differentiable at any point in $[0, 1]$. Its derivative is a reduced-order Bézier curve with the form of

$$f'(s) = k \sum_{r=0}^{k-1} B_r^{k-1}(s)(p_{r+1} - p_r)$$

which is also differentiable. Therefore, a Bézier curve of degree $k$ has derivatives of all orders.

 Quadratic and cubic Bézier curves are most common in practice, where $k = 2$ and $3$ respectively. Particularly for the cubic Bézier curve, Eq. (4) has the matrix form of

$$f(s) = \mathbf{P} \mathbf{M} z,$$

where

$$\mathbf{P} = (p_0, p_1, p_2, p_3)$$
defined the total order by Definition 1 associated with the self-dual proper cone $\mathbf{R}_d^+ = \{ \rho : \rho^T \mathbf{x} \geq 0, \forall \mathbf{x} \in \mathbf{R}_d^+ \} [40]$. 

Definition 1. The order between two vectors $\mathbf{x}$ and $\mathbf{y}$ in the subset $\mathcal{S} \subseteq \mathbf{R}^d$ is defined to be 

$$\mathbf{x} \preceq \mathbf{y} \iff \begin{pmatrix} \delta_1(y_1 - x_1) \\ \delta_2(y_2 - x_2) \\ \vdots \\ \delta_d(y_d - x_d) \end{pmatrix} \in \mathbf{R}_d^+$$

(13)

where $\mathbf{x} = (x_1, x_2, \ldots, x_d)^T, \mathbf{y} = (y_1, y_2, \ldots, y_d)^T$, 

$$\delta_j = \begin{cases} 1, & j \in \mathbf{E} \\ -1, & j \in \mathbf{F} \end{cases},$$

(14)

$$\mathbf{E} \cap \mathbf{F} = \emptyset \text{ and } \mathbf{E} \cup \mathbf{F} = \{1, 2, \ldots, d\}.$$ 

For a given ranking task, $\mathbf{x}$ precedes $\mathbf{y}$ for $x_j < y_j (j \in \mathbf{E})$ and $x_j > y_j (j \in \mathbf{F})$. Let 

$$\alpha = (\delta_1, \delta_2, \ldots, \delta_d)^T$$

(15)

and name it monotonicity index. For example, $\alpha = (1, 1, -1, -1)^T$ for a ranking problem, $\mathbf{E} = \{1, 2\}$ and $\mathbf{F} = \{3, 4\}$ accordingly. For any pair of ranking candidates $\mathbf{x}$ and $\mathbf{y}$, $x_j < y_j$ holds for $j = 1, 2$ while $x_j > y_j$ holds for $j = 3, 4$. Therefore, the full ranking list $\mathbf{x}_1 \preceq \mathbf{x}_2 \preceq \cdots \preceq \mathbf{x}_n$ is in the same order of the increasing orders of the first and second attributes observations while also in the same order of the decreasing orders of the third and fourth attributes observations. $\alpha$ is unique for one given ranking task and varies from task to task. It can be the prior knowledge along with attributes, or can be learned from data points $\mathbf{X}$.

As $\mathbf{R}$ is totally ordered, we prefer to grade each point with a real value to help with ranking. Assume $\varphi : \mathbf{R}^d \rightarrow \mathbf{R}$ is the ranking function to assign a score which provides the ordering of $\mathbf{x}$. $\varphi$ is required to be order-preserving so that $\varphi(\mathbf{x})$ has the same ordering in $\mathbf{R}$ as $\mathbf{x}$ in $\mathbf{R}^d$. In order theory, an order-preserving function is also called isotone or monotone [41].

Definition 2 ([41]). A function $\varphi : \mathbf{R}^d \rightarrow \mathbf{R}$ is called monotone (or, alternatively, order-preserving) if 

$$\mathbf{x} \preceq \mathbf{y} \implies \varphi(\mathbf{x}) \leq \varphi(\mathbf{y})$$

(16)

and strictly monotone if 

$$\mathbf{x} \preceq \mathbf{y}, \mathbf{x} \neq \mathbf{y} \implies \varphi(\mathbf{x}) < \varphi(\mathbf{y})$$

(17)

Order-preserving is the basic requirement for a ranking function. For a partially ordered set, $\varphi$ should assign a score to $\mathbf{x}$ no more than the score to $\mathbf{y}$ if $\mathbf{x} \preceq \mathbf{y}$. Moreover, if $\mathbf{x} \neq \mathbf{y}$ also holds, the score assigned to $\mathbf{x}$ must be smaller than the score to $\mathbf{y}$. As $\mathcal{S}$ is totally ordered and different points should be assigned with different scores, the ranking function is required to be strictly monotone as stated by Eq. (17).

Example 2. In addition to the two indicators in Example 1, another two indicators are taken to evaluate life qualities of countries: IMR$^3$ and Tub$^4$. It is easily known that the

3. Infant Mortality Rate per 1000 born.
4. New cases of infectious Tuberculosis per 100,000 of population.
life quality of one country would be higher if it has a higher LEB and GDP while a lower IMR and Tub. Let numerical observations on four countries to be $x_i = (2.1, 62.7, 75.59), x_M = (11.3, 75.5.12.30), x_G = (32.1, 79.2, 6.4), \text{and } x_N = (47.6, 80.1, 3.3)$ respectively. By Eq. (13), they have the ordering $x_M \succeq x_M \succeq x_G \succeq x_N$ with monotonicity index $\alpha = (1, 1, -1, -1)^T$, where $E = \{1, 2\}$ and $F = \{3, 4\}$. $\alpha$ also discovers the relationship between indicators for ranking. GDP is in the same direction with LEB, but in the opposite direction with IMR and Tub. Let $\varphi(x_i) = 0.407, \varphi(x_M) = 0.593, \varphi(x_G) = 0.785$ and $\varphi(x_N) = 0.891$. Then $\varphi$ is a strictly monotone mapping which strictly preserves the ordering in $\mathbb{R}^d$.

**Definition 3 ([42]).** For a non-negative integer $h$, $\varphi(x)$ is of the differentiability class $\mathcal{C}^h$ if all the partial derivatives $\frac{\partial^h \varphi}{\partial x_1^k \partial x_2^j \ldots \partial x_d^l}$ exist and are continuous for all $x$ in $\text{dom}\varphi$, where $k$ is a non-negative integer, $i_1, i_2, \ldots, i_k \in \{1, 2, \ldots, d\}$ and $h_1, h_2, \ldots, h_k \in \{0, 1, \ldots, h\}$ are two integer sequences, and $h_1 + h_2 + \ldots + h_k = h$.

**Theorem 1 ([40]).** Let $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}$ be of the differentiability class $\mathcal{C}^h (h \geq 1)$, $\varphi$ is monotone if and only if

$$\nabla \varphi(x) \succcurlyeq 0 \quad (18)$$

where $0$ is the zero vector. $\varphi$ is strictly monotone if

$$\nabla \varphi(x) \succ 0 \quad (19)$$

Theorem 1 provides first-order conditions for monotonicity. Note that ‘$\succ$’ denotes a strict partial order [40]. Let

$$\nabla \varphi(x) = \left( \frac{\partial \varphi}{\partial x_1}, \frac{\partial \varphi}{\partial x_2}, \ldots, \frac{\partial \varphi}{\partial x_d} \right)^T. \quad (20)$$

$\nabla \varphi(x) \succ 0$ infers $\frac{\partial \varphi}{\partial x_j} > 0$ for $j \in E$ and $\frac{\partial \varphi}{\partial x_j} < 0$ for $j \in F$. $\nabla \varphi(x) \succ 0$ infers that each component of $\nabla \varphi(x)$ does not equal to zero. By the case of strict monotonicity in Theorem 1, $\nabla \varphi(x) \succ 0$ infers not only that $\varphi$ is strictly monotone from $\mathbb{R}^d$ to $\mathbb{R}$, but also that the value $s = \varphi(x)$ is increasing with respect to $x_j (j \in E)$ and decreasing with respect to $x_j (j \in F)$. Vice versa, if $\frac{\partial \varphi}{\partial x_j}$ is bigger than zero for $j \in E$ and smaller than zero for $j \in F$, $\nabla \varphi(x) \succ 0$ holds and infers $\varphi$ is a strictly monotone mapping. Lemma 1 can be concluded immediately.

**Lemma 1.** $s = \varphi(x)$ is strictly monotone if and only if $s$ is strictly monotone along $x_i$ with fixed the others $x_j (j \neq i)$.

Further more, a strictly monotone mapping infers a one-to-one mapping that for a value $s \in \text{rang}(\varphi)$ there is exactly one point $x \in \text{dom}(\varphi)$ such that $\varphi(x) = s$. If the point $x$ is denoted by $x = f(s)$, $f : \mathbb{R} \rightarrow \mathbb{R}^d$ is called the inverse mapping of $\varphi$ and inherits the property of strict monotonicity of its origin $\varphi$.

**Theorem 2.** Assume $\nabla \varphi(x) \succ 0$. There exists an inverse mapping denoted by $f : \text{rang}(\varphi) \rightarrow \text{dom}(\varphi)$ such that $\nabla f(s) \succ 0$ holds for all $s \in \text{rang}(\varphi)$, that is for $s_1, s_2 \in \text{rang}(\varphi)$

$$s_1 < s_2 \implies f(s_1) \leq f(s_2), \quad f(s_1) \neq f(s_2). \quad (21)$$

Proof of Theorem 2 can be found in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2015.2441692. The theorem also holds in the other direction. Assuming $f : \mathbb{R} \rightarrow \mathbb{R}^d$, if $\nabla f(s) \succ 0$, there exists an inverse mapping $\varphi : \text{rang}(f) \rightarrow \text{dom}(f)$ and $\nabla \varphi(x) \succ 0$ holds for all $x \in \text{rang}(f)$. Because of the one-to-one correspondence, $f$ and $\varphi$ share the same geometric properties such as scale and translation invariance, smoothness and strict monotonicity [42].

**4 Meta-Rules**

Motivated by PageRank [9], we draw five essential items from ranking knowledge, named meta-rules, to serve as high-level assessments for unsupervised ranking results. These meta-rules can also guide the design of ranking functions. Assume a ranking function $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}$, where $\varphi(x)$ outputs a real value $s = \varphi(x)$ as the ranking score for a given point $x$. Five essential meta-rules are formalized to be scale and translation invariance, strict monotonicity, compatibility of linearity and nonlinearity, smoothness, and explicitness of parameter size.

**4.1 Scale and Translation Invariance**

**Definition 4 ([43]).** A ranking rule is invariant to scale and translation if for $x \preceq y$

$$\varphi(x) \preceq \varphi(y) \iff \varphi(L(x)) \preceq \varphi(L(y)). \quad (22)$$

where $L(\cdot)$ performs scale and translation.

This is a basic requirement for a ranking function and has been referred in multiple literatures (e.g., [7], [43]). In general, numerical observations on different indicators are taken on different quantity dimensions. In Example 1, GDP is measured in thousands of dollars while LEB ranges from 40 to 90 years. They are not in the same quantity dimensions. Preprocessing of datasets needs to be done according to normalization or standardization, both of which takes the transformation of scale and translation, while original order is required to be preserved. If let $L$ be a linear transformation on $\mathbb{R}^d$, we have $x \preceq y \iff L(x) \preceq L(y)$ for $x, y \in \mathbb{R}^d$ [43]. Therefore, a ranking function $\varphi(x)$ should produce the same ranking list before and after transformation of scaling and translating.

**4.2 Strict Monotonicity**

**Definition 5 ([41]).** $\varphi(x)$ is strictly monotone if $\varphi(x_i) < \varphi(x_j)$ for $x_i \preceq x_j$ and $x_i \neq x_j (i \neq j)$.

Monotonicity in Definition 2 is specified here as one of meta-rules for ranking. It has been widely used as ranking knowledge constraints in ordinal classification and regression [11], [36], [37]. Monotonicity helps improve these supervised tasks greatly by embedding it into learning models. For unsupervised ranking problems, ranking functions with monotonicity constraint are also expected to produce ranking lists closer to “ground-truth” than ranking functions without it. $\varphi(x_i) = \varphi(x_j)$ holds if and only if $x_i = x_j (i \neq j)$ in Example 1 (Fig. 2), $x_1 \preceq x_2$ and $x_i \neq x_j$ indicate that a higher score should be assigned to $x_3$ than $x_1$. And so do $x_3$ and $x_1$. Therefore, the ranking function $\varphi(x)$ is required to be a strictly monotone mapping. $\varphi$ in Example 2 is to the point referred here.
Generally, nonlinear regression will introduce the model selection problem. However, conflicting results would occur with conventional methods of model selection. For example, the Bayesian information criterion (BIC) for model selection is given by [25]

\[ BIC = \text{Error} + \text{ModelSize} \]  

and there are two models A and B with:

\[ BIC_A = 10 = 2 + 8 \]  
\[ BIC_B = 9 = 6 + 3. \]

With the BIC criterion, the model B is preferred, in conflict with the result with error criterion that the model A is preferred. To avoid this conflict, the explicitness of parameter size ensures a fair comparison between models of the same parameter size.

5 RANKING PRINCIPAL CURVES

For unsupervised ranking, we present a RPC model which follows all the five meta-rules. The RPC is parametrically designed with a cubic Bézier curve of strict monotonicity.

5.1 RPC Model

Example 4. As we know, the life quality of one country would be higher if it has a higher LEB and GDP. Let numerical observations of two countries be \( x_c = (78, 12) \) and \( x_P = (72, 21) \). One would get the order of \( x_P \leq x_c \) with LEB and \( x_P \leq x_c \) with GDP. It is a very often phenomenon that two objects get contrary orders on different indicators.

In real applications, the order of precedence cannot be directly determined according to comparison dimension by dimension due to the fact that multi-attribute objects would show different rankings orders for different indicators. In modeling, we assume that the contrary was incurred by noise in numerical observations of objects. That is, \( x = x_{true} + \varepsilon \), where \( \varepsilon \) is the noise to \( x_{true} \) which is the noiseless observation and \( x \) is the observed data. The score \( s \) should be produced for \( x_{true} = x - \varepsilon \) by a ranking function, that is \( s = \varphi(x - \varepsilon, \theta) \). As a ranking function, \( \varphi \) is assumed to be strictly monotone. Thus by Theorem 2, there exists an inverse function \( f \) for \( \varphi \) such that

\[ x = f(s, \theta) + \varepsilon \]

which is the principal curve model [26]. The inverse function can be taken as the generating function for numerical observations of ranking candidates from the score \( s \) which can be regarded to be pre-existing. The principal curve is used to approximate the latent generating function. The design of a ranking function turns to find the corresponding principal curve for the given numerical observations.

Correspondingly, five meta-rules for ranking functions \( \varphi \) should be also followed by the principal curve \( f \) for ranking. However, general principal curve models are probably not suitable to carry out ranking tasks. As the degeneration of the principal curve, the first PCA [17] can well depict the ranking skeletons of datasets like ellipses but fail for the cases of crescents (e.g., Fig. 5a). Polyline approximations [27] of the principal curve might go against smoothness and
strict monotonicity (e.g., Fig. 5b). A smooth principal curve model (e.g., [28]) would still go against strict monotonicity (e.g., Fig. 5c). Going against smoothness and strict monotonicity might bring problems which are illustrated in Example 1 (Fig. 2) and Example 3 (Fig. 4). Within the framework of Fig. 1, all the five meta-rules can be modeled as constraints to the ranking function. Since a principal curve is defined to be smooth and invariant to scale and translation [26], the constraint of strict monotonicity would make it be capable of performing ranking tasks (e.g., Fig. 5d). Naturally, the principal curve should have a known parameter size for a fair comparison. We present Definition 9 for unsupervised ranking with a principal curve.

**Definition 9.** A one-dimensional curve \( f(s, \theta) \) in \( d \)-dimensional space is called a RPC if \( f(s, \theta) \) is a strictly monotone principal curve of given data cloud and it is explicitly expressed with known parameters \( \theta \) of limited size.

### 5.2 RPC Formulation with Bézier Curves

Remembering attractive properties of the cubic Bézier curve (Section 2.2), we select the cubic Bézier curve with constraints on control points to parametrically model an RPC. Specifically, the RPC is modeled to be

\[
\begin{align*}
\mathbf{f}(s) &= \sum_{i=0}^{3} B_i^3(s) \mathbf{p}_i, \quad s \in [0, 1],
\end{align*}
\]

where control points are RPC parameters which is of size \( 4d \). Moreover, we specify four propositions for the cubic Bézier curve following the other four meta-rules.

**Proposition 1.** A cubic Bézier curve is invariant to scale and translation.

**Proof.** Specifying the affine transformation \( \mathbf{L} \) to be scale and translation, \( \mathbf{L} \) can be imposed directly on control points without changing the score \( s \). That is

\[
\Lambda \mathbf{f}(s) + \mathbf{b} = \mathbf{LPMz} + \mathbf{b} = (\mathbf{AP} + \mathbf{b}) \mathbf{Mz},
\]

where \( \Lambda \) is a diagonal matrix with scaling factors to dimensions and \( \mathbf{b} \) is the translation vector.

This proposition allows data preprocessing of regularization without changing ranking scores. In this work, we regularize data into \([0, 1]^d\).

**Proposition 2.** \( f(s) \) is strictly monotone for \( s \in [0, 1] \) with \( p_0 = \frac{1}{2}(1 - \alpha), p_3 = \frac{1}{2}(1 + \alpha) \) and \( p_1, p_2 \in (0, 1)^d \).

**Proof.** Let end points are denoted by \( p_0 = \frac{1}{2}(1 - \alpha) \) and \( p_3 = \frac{1}{2}(1 + \alpha) \). Control points \( p_1 \) and \( p_2 \) are the determinants for monotonic shapes of the cubic Bézier curves (Fig. 3). It has been proved [35] that \( \nabla f(s) \geq 0 \) for \( p_1, p_2 \in [0, 1]^d \) and \( \nabla f(s) = 0 \) occurs in the border of \([0, 1]^d\). When we restrict \( p_1 \) and \( p_2 \) in the interior of \([0, 1]^d\), that is \((0, 1)^d\), we get \( \nabla f(s) > 0 \) which infers strict monotonicity by Theorem 1.

**Proposition 3.** A cubic Bézier curve has the capabilities of linearity and nonlinearity.

**Proof.** Eq. (27) is a nonlinear function of \( s \) due to \( B_i^3(s) \) (Eq. (5)). If \( p_0, p_1, p_2, p_3 \) are evenly spaced (i.e., form an arithmetic progression), the cubic Bézier curve becomes a linear function \( f(s) = p_0 + s(p_1 - p_0) \) which runs from \( p_0 \) to \( p_1 \) as \( s \) runs from 0 to 1.

**Proposition 4.** A cubic Bézier curve is of the differentiability class \( C^3 \).

Proposition 4 is a directly results of the property of reduced-order derivatives in Section 2.2. What is the most important, there always exists an RPC parameterized by a cubic Bézier curve of strict monotonicity. This proposition allows data preprocessing of regularization without changing ranking scores. In this work, we regularize data into \([0, 1]^d\).

**Theorem 3.** Assume that \( x \) is the numerical observation of a ranking candidate and that \( E ||x||^2 < \infty \). There exists \( \mathbf{P}^* \in [0, 1]^d \) such that \( \mathbf{f}^*(s) = \mathbf{P}^*\mathbf{Mz} \) is strictly monotone and

\[
J(\mathbf{P}^*) = \inf \left\{ J(\mathbf{P}) = E \left( \inf_{s} ||\mathbf{x} - \mathbf{P}\mathbf{Mz}||^2 \right) \right\}.
\]

Proof of Theorem 3 can be found in Appendix B, available online.

### 6 RPC Learning Algorithm

To perform unsupervised ranking from the multi-attribute numerical observations of ranking candidates \( \mathbf{X} = (x_1, x_2, \ldots, x_n) \), an RPC modeled by a cubic Bézier curve should be learned to achieve the infimum of the estimation of \( J(\mathbf{P}) \) in Eq. (29). By the principal curve definition proposed by Hastie and Stuetzle[26], the RPC is the curve which minimizes the summed residual \( \varepsilon \). Therefore, the ranking task is formulated as a nonlinear optimization problem

\[
\min_{\mathbf{P}, \mathbf{s}} \quad J(\mathbf{P}, \mathbf{s}) = \sum_{i=1}^{n} ||x_i - \mathbf{P}\mathbf{Mz}_i||^2
\]

s.t.

\[
\left( \frac{\partial \mathbf{P}\mathbf{Mz}}{\partial s} \right)_{s=s_i}^T (x_i - \mathbf{P}\mathbf{Mz}) = 0,
\]

\[
\mathbf{s} = (s_1, s_2, \ldots, s_n), \quad \mathbf{z}_i = (1, s_i, s_i^2, s_i^3)^T,
\]

\[
\mathbf{P} \in [0, 1]^{4d}, \quad s_i \in [0, 1],
\]

\[
i = 1, 2, \ldots, n,
\]
where Eq. (31) determines $s_i$ to find the point on the curve which has the minimum residual to reconstruct $x_i$ by $f(s_i)$. Obviously, a local minimizer $(P^*, s^*)$ can be achieved in an alternating minimization way

$$P^{(t+1)} = \arg \min_P \sum_{i=1}^{n} \left\| x_i - PMZ_i^{(t)} \right\|^2$$

(32)

$$\left( \frac{\partial P^{(t+1)}Mz}{\partial s} \right)^T \left( x_i - P^{(t+1)}Mz \right) \bigg|_{s=s_i^{(t)}} = 0$$

(33)

where $t$ means the $t$th iteration.

The optimal solution of Eq. (32) has an explicit expression. Associate $X$ with $Z$

$$Z = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ s_1 & s_2 & \cdots & s_n \\ s_1' & s_2' & \cdots & s_n' \end{pmatrix} = (z_1, z_2, \ldots, z_n)$$

(34)

and Eq. (30) can be rewritten in matrix form

$$J(P, s) = \|X - PMZ\|_F^2$$

$$= tr(X^TX) - 2tr(PMZ^T)$$

$$+ tr(PMZ^TMP^T).$$

(35)

Setting the derivative of $J$ with respect to $P$ to zero

$$\frac{\partial J}{\partial P} = 2\left(P(MZ)(MZ)^T - X(MZ)^T\right) = 0$$

(36)

and remembering $A^+ = A^T(AA^T)^+$, we get an explicitly expression for the minimum point of Eq. (30)

$$P = X(MZ)^T \left( (MZ)(MZ)^T \right)^+ = X(MZ)^+$$

(37)

where $(\cdot)^+$ takes pseudo-inverse computation. Based on the $t$th iterative results $Z^{(t)}$, the optimal solution can be given by substituting $Z^{(t)}$ into Eq. (37) which is $P^{(t+1)} = X(MZ^{(t)})^+$. However, $(MZ^{(t)})^+$ is computationally expensive in numerical experiments and $X$ is always an ill-conditioned matrix which has a high condition number, resulting in a very small change in $Z^{(t)}$ would produce a tremendous change in $P^{(t+1)}$. $Z^{(t)}$ is not the optimal solution of Eq. (30) but a intermediate result of the iteration, and $P^{(t+1)}$ would thereby go even far away from the optimal solution. To settle out the problem, we employ the Richardson iteration [38] with a preconditioner $D$ which is a diagonal matrix with the $L_2$ norm of columns of $(MZ^{(t)})(MZ^{(t)})^T$ (Eq. (37)) as its diagonal elements. Then $P^{(t+1)}$ is updated iteratively according to

$$P^{(t+1)} = P^{(t)} - \gamma^{(t)}(P^{(t)}(MZ)^T(MZ)^T)$$

$$- X(MZ^{(t)})^TP^{(t)}$$

(38)

where $\gamma^{(t)}$ is a scalar parameter such that the sequence $P^{(t)}$ converges. In practice, we set

$$\gamma^{(t)} = \frac{2}{\lambda_{min}^{(t)} + \lambda_{max}^{(t)}}$$

(39)

where $\lambda_{min}^{(t)}$ and $\lambda_{max}^{(t)}$ is the minimum and maximum eigenvalues of $(MZ^{(t)})(MZ^{(t)})^T$ respectively [50]. After getting $P^{(t+1)}$, the score vector $s^{(t+1)}$ can be calculated as the solution to Eq. (33). Eq. (33) is a quintic polynomial equation which rarely has explicitly expressed roots.

Algorithm 1. Algorithm to learn an RPC.

Input:

- $X$: data matrix;
- $\xi$: a small positive value;

Output:

- $P^*$: control points of the learned Bézier curve
- $s$: the score vector of objects in the set.

1: Initialize $P^{(0)}$;
2: while $\Delta J > \xi$ do
3: Do GSS to find the approximate solution $s^{(t)}$;
4: Compute $P^{(t+1)}$ using a preconditioner;
5: if $\Delta J < 0$ then
6: break;
7: end if
8: end while

Algorithm 1 summarizes the alternative optimization procedure. Before performing the ranking task, numerical observations of objects should be normalized into $[0, 1]^d$ by

$$\hat{x} = \frac{x - x_{min}}{x_{max} - x_{min}},$$

(40)

where $\hat{x}$ is the normalized vector of $x$, $x_{min}$ the minimum vector and $x_{max}$ the maximum vector. Grading scores would be unchanged as scaling and translating are only performed on control points and end points (Eq. (28)) without changing the interpolation values. In Step 2, we initialize the end points as $p_0 = \frac{1}{2}(1 - \alpha)$ and $p_3 = \frac{1}{2}(1 + \alpha)$, and randomly select samples as control points. During learning procedure, $P^{(t)}$ is automatically learned to make a Bézier curve be an RPC. In Step 6, $\Delta J < 0$ occurs when $J$ begins to increase. In this case, the algorithm stops updating $(P^{(t)}, s^{(t)})$ and gets a local minimum $J$. Proposition 5 guarantees the convergency of the sequence found by RPC learning algorithm (proof can be found in Appendix C, available online). Therefore, the RPC learning algorithm finds a converging sequence of $(P^{(0)}, s^{(0)})$ to achieve the infimum in Eq. (29).

Proposition 5. If $P^{(t)} \to P^*$ as $t \to \infty$, $J(P^{(t)}, s^{(t)})$ is a decaying sequence which converges to $J(P^*, s^*)$ as $t \to \infty$.

Algorithm 1 converges in limited steps. In each step, $P$ is updated in $4 \times d$ size and scores for points are calculated in $n$ size. When iteration stops, ranking scores are produced along with $P$. In summary, the computational complexity of RPC unsupervised ranking model is $O(n)$.

RPC learning algorithm learns a ranking function in a completely different way from traditional ranking methods. The ranking function is in constraints of five ranking meta-rules. Integrating meta-rules with ranking functions makes
the ranking rule be more in line with human knowledge about ranking problems. As a high level knowledge, these meta-rules are capable of evaluating ranking performance. Moreover, ranking is carried out following the principle of unsupervised ranking, "let the data speak for themselves". For unsupervised ranking, the structure of the dataset contains the ordinal information between objects. Ideally, the RPC can thread through all the objects successively. Yet in practice, the most influential indicators are selected to estimate the order of objects and noise exists. In the case we know nothing about the rest factors, we would better to minimize the effect which we formulate to be error ε. Therefore, minimizing errors is adopted as the learning objective.

7 EXPERIMENTS

Since no ground-truth ranking labels are available, quantitative evaluations can be hardly carried out. In this part, RPC is compared with two heuristic rank aggregation methods and two unsupervised ranking methods on five meta-rules in the sense of meta-rules and artificial data. Two real applications, world university ranking and journal ranking, are carried out to validate RPC’s applicability.

7.1 Comparisons on Meta-Rules

There have been several simple rank aggregation methods [22, 46] and unsupervised ranking strategies have been developed recently, most of which focus on the information retrieval of web search [3, 19, 21, 22, 23]. Ranking for web search is query-oriented where websites are ranked by some strategies with the referenced query [3]. For ranking tasks, some researchers prefer to aggregate different ranking lists of the same set of objects in order to get a consensus order. We compare RPC with two heuristic rank aggregation methods and two unsupervised ranking methods and summarize comparisons on meta-rules in Table 3. Computation complexities and parameter sizes are summarized in Table 4.

7.1.1 Comparisons with Rank Aggregation Methods

For simple aggregation methods, we select Borda count (BC) [22] and CombSUM [46] for comparisons. As an order-based method, BC cannot efficiently detect ordinal information embedded in the numerical observations due to its non-smoothness, illustrated in Table 2. BC provides the same order when A in Table 2(a) is changed to A0 in Table 2(b). Furthermore, BC implies nonlinear relationships between indicators and ranking scores, resulting in a model bias. As a score-based method, CombSUM provides different scores for A and A0 but still the same ranking lists. Whether score-based or order-based, the simple aggregation methods apply the same aggregation strategy for all cases and do not consider the variability of different ranking tasks.

Among several unsupervised ranking aggregation methods, MPM is the latest developed method [21]. It shows very good performance on aggregating meta-search results in the sense of NDCG and MAP on LETOR [13]. MPM has an implicit (s = ψ(x), ψ is implicit) ranking function and can deal with partial lists such that MPM has wide applications on supervised and unsupervised ranking. In contrast, RPC

| Meta-Rule                          | BC | CombSUM | MPM | Elmap | RPC |
|-----------------------------------|----|---------|-----|-------|-----|
| Scale and Translation Invariance  | Y  | N       | Y   | Y     | Y   |
| Strict Monotonicity               | Y  | Y       | U   | N     | Y   |
| Capacities of Linearity and Nonlinearity | N  | N       | Y   | Y     | Y   |
| Smoothness                        | N  | C³      | U   | C³    | C³ |
| Explicitness of Parameter Size    | Y  | Y       | Y   | Y     | Y   |

TABLE 3

Meta-Rules Followed by Different Ranking Methods

(Y - Yes, N - No, U - Unknown)

| Complexity | MPM | Elmap | RPC |
|------------|-----|-------|-----|
|             | O(dn) | O(n²/3) | O(n) |
| Parameter Size | n | k(2 + d)² | 4d |

1 MPM ranked lists are in their inverse orders (MPM Package is available on http://www.cs.toronto.edu/~mvolkovs/).
is an explicit ranking function and requires full lists of attributes such that RPC is unsuitable for aggregating search results of different engines (like LETOR). However, the competitive advantage of RPC lies in visualization such that the user can have a clear scope of the “ranking coordinate”. The parameterized RPC makes it easier to interpret the ranking rules for ranking candidates.

7.1.2 Comparison with Elmap

RPC is motivated from Elmap [28] yet makes the principal curve application in ranking more systematical. Compared to Elmap, RPC takes the advantages of smoothness and strict monotonicity which have been widely used in ordinal regression and classification. Figs. 2 and 4 show the necessities of these meta-rules for principal curves to perform ranking tasks. For the synthetical data in Table 2, Elmap produces the same score for $A_0$ and $B$ due to the non-smoothness (with Elmap Package available on http://www.ihes.fr/~zinovyev/vida/elmap/index.htm) although it also detects the numerical information. They are compared also in the sense of fitting MSE as principal curves for numerical observations of ranking candidates (Fig. 6). As Elmap produces different principal curves for different nodes number assignments, we compare two cases of nodes number settings: Elmap with 4 points (Elmap$_4$, the same parameter numbers with RPC), and Elmap with auto selected points (Elmap$_{Auto}$). In the sense of fitting MSE, RPC obtains lower residual than both Elmap$_4$ and Elmap$_{Auto}$.

7.2 Applications

Unsupervised ranking of multi-attribute objects has a widely applications. Taking the journal ranking task for illustration, there have been many indices to rank journals, such as impact factor (IF) [51] and Eigenfactor [52]. Different indices reflect different aspects of journals and provide different ranking lists for journals. Different aggregating methods may produce similar ranking results, but the ranking list of the method which matches the prior information the most will be preferred by users. RPC model is proposed as a new ranking framework following the prior information which provides an ordering along the “ranking skeleton” of data distribution.

In this paper, we perform ranking tasks with RPCs to produce comprehensive evaluations on two open access datasets of countries and journals with the open source software Scilab (5.4.1 version) on a Ubuntu 12.04 system with 4 GB memory. Due to space limitation, we just list parts of their ranking lists. The whole lists are available in the supplemental materials. We also carry out ranking on universities, the ranking list of which is also available in the supplemental materials.

7.2.1 Results on Life Qualities of Countries

Zinovyev and Gorban [12] ranked 171 countries by life qualities of countries with data driven from GAPMINDER (http://www.gapminder.org/) based on four indicators as in Example 2. For comparison, we use the same four GAPMINDER indicators as in [12]. The RPC learned by Algorithm 1 is shown in two-dimensional visualizations in Fig. 7 and part of the ranking list is illustrated in Table 5.

RPC portrays the development trends, based on the noised observations, with different shapes, including linearity and nonlinearity (Fig. 7). $\alpha = [1, 1, -1, -1]^T$ (just as Example 2) indicates the monotonicity directions for indicators. In the beginning, a small amount of GDP increasing brings about tremendous increasing of LEB and tremendous decreasing of IMR and Tub. When GDP exceeds $14300 (0.2 as normalized value in Fig. 7) per person, increasing GDP does result in little LEB increase, so does IMR and Tub decrease. As a matter of fact, it is hard to improve further LEB, IMR and Tub when they are close to the limit of human evolution. When control points (provided in the bottom of Table 5) have overlap values in one dimension (e.g., GDP), it means this dimension is a “C”-shape curve. Otherwise, the dimension is a “S”-shape curve. RPC scores of countries are nonlinear interpolation values between the best country (Luxembourg) and the worst country (Swaziland) with the control of $p_1$ and $p_2$ to determine their specific developing trends.
7.2.2 Results on Journal Ranking

We also apply RPC model to rank journals with data accessible from the Web of Knowledge (http://wokinfo.com/) which is affiliated to Thomson Reuters. Thomson Reuters publishes annually journal citation reports (JCR) which provide information about academic journals in the sciences and social sciences. JCR2012 reports citation information with indicators of impact factor, 5-year impact factor, Immediacy Index, Eigenfactor Score, and Article Influence Score. After journals with data missing are removed from the data table (58 out of 451), RPC model tries to provide a comprehensive ranking list of journals in the categories of computer science: artificial intelligence, cybernetics, information systems, interdisciplinary applications, software engineering, theory and methods. Table 6 gives part of the ranking list produced by RPC model based on JCR2012. Two-dimensional visualization of the RPC can be found in supplemental materials.

For this ranking task, a journal will rank higher with a higher value for each indicator, that is $\alpha = [1, 1, 1, 1, 1]$. From Table 6, IEEE Transactions on Knowledge and Data Engineering (TKDE) is ranked in a higher place than IEEE Transactions on Systems, Man, and Cybernetics-Part A (SMCA) although SMCA has a higher IF (2.183) than TKDE (1.892). The lower influence score (0.767) of SMCA brings it down the ranking list (vs. 1.129 for TKDE). Therefore, TKDE gets a higher comprehensive evaluating score and wins a higher ranking place in the ranking list. This means that one indicator does not tell the whole story of ranking lists. RPC produces a ranking list for journals taking into account several indicators of different aspects.

### Table 6

| Country       | GDP   | LEB   | IMR   | Tub   | Elmap [12] Score | Order | Score | Order | RPC Score | Order |
|---------------|-------|-------|-------|-------|------------------|-------|-------|-------|-----------|-------|
| Luxembourg    | 70014 | 79.56 | 6     | 4     | 0.892            | 1     | 1.000 | 1     | 1         | 1     |
| Norway        | 47551 | 80.29 | 3     | 3     | 0.647            | 2     | 0.872 | 2     | 1         | 1     |
| Kuwait        | 44947 | 77.258| 11    | 10    | 0.608            | 3     | 0.848 | 3     | 1         | 1     |
| Singapore     | 41479 | 79.627| 12    | 2     | 0.578            | 4     | 0.830 | 4     | 1         | 1     |
| United States | 41674 | 77.93 | 2     | 7     | 0.575            | 5     | 0.827 | 5     | 1         | 1     |
| Turkey        | 7786  | 71.396| 13    | 26    | 0.090            | 76    | 0.629 | 75    | 1         | 1     |
| Iran          | 10692 | 70.618| 11    | 31    | 0.105            | 69    | 0.624 | 76    | 1         | 1     |
| Armenia       | 3903  | 73.129| 32    | 23    | 0.074            | 78    | 0.624 | 77    | 1         | 1     |
| China         | 4909  | 72.555| 45    | 21    | 0.079            | 77    | 0.622 | 78    | 1         | 1     |
| Samoa         | 4872  | 70.807| 9     | 24    | 0.070            | 81    | 0.618 | 79    | 1         | 1     |
| $p_0$         | 44713 | 61.218| 2     | 0     | -                | -     | -     | -     | -         | -     |
| $p_1$         | 330   | 80.4  | 2     | 0     | -                | -     | -     | -     | -         | -     |
| $p_2$         | 330   | 59.7  | 33    | 43    | -                | -     | -     | -     | -         | -     |
| $p_3$         | 1581.624 | 41.68 | 290   | 151   | -                | -     | -     | -     | -         | -     |

| Title                  | Impact Factor (IF) | 5-Year IF | Immediacy Index | Eigenfactor | Influence Score | RPC |
|------------------------|-------------------|-----------|-----------------|-------------|-----------------|-----|
| IEEE T PATTERN ANAL    | 4.795             | 7         | 6.144           | 0.625       | 0.05237         | 3   |
| ENTERP INF SYST UK     | 9.256             | 1         | 4.771           | 2.682       | 0.00173         | 230 |
| J STAT SOFTW           | 4.910             | 4         | 5.907           | 0.753       | 0.01744         | 20  |
| MIS QUART              | 4.659             | 8         | 7.474           | 0.705       | 0.01036         | 49  |
| ACM COMPUT SURV        | 3.543             | 21        | 7.854           | 0.421       | 0.00640         | 80  |
| DECIS SUPPORT SYST     | 2.201             | 51        | 3.037           | 0.196       | 0.00994         | 52  |
| COMPUT STAT DATA AN    | 1.304             | 156       | 1.449           | 0.415       | 0.02601         | 11  |
| IEEE T KNOWL DATA EN   | 1.892             | 82        | 2.426           | 0.217       | 0.01256         | 37  |
| MACH LEARN             | 1.467             | 133       | 2.143           | 0.373       | 0.00838         | 81  |
| IEEE T SYST MAN CY A   | 2.183             | 53        | 2.44            | 0.465       | 0.00728         | 69  |
|                        |                   |           |                 |             |                 |     |
the most preferred. For the models which satisfy all the meta-rules and need further evaluation, we will apply a fitting criterion, such as MSE (Fig. 6), to determine the final model.

8 Conclusions

Ranking and its tools have and will have an increasing impact on the behavior of human, either positively or negatively. However, those ranking activities are still facing many challenges which have greatly restrained to the rational design and utilization of ranking tools. Generally, ranking in practice is an unsupervised task which encounters a critical challenge that there is no ground truth to evaluate the provided lists.

Motivated by [9], [10], [11], this work presents five essential meta-rules (there might be other meta-rules to study on) as a high-level evaluation for unsupervised ranking performance. They are scale and translation invariance, strict monotonicity, linear/nonlinear compatibility, smoothness and explicitness of parameter size. They can also guide the design of ranking functions as constraints. Furthermore, we propose a RPC model which meets all the five meta-rules. Parametrically, RPC is modeled with a cubic Bézier curve by restricting control points in the interior of the hypercube $[0, 1]^d$, which can be learned from the data distribution without human interventions. Experiments on life qualities of countries and journals of computer sciences show that the proposed RPC model is applicable in practice.

However, RPC model requires full lists of attributes such that it cannot deal with missing value fields and partial lists. Considering that it is common in practice, we will endeavor on this problem in our future works. In application, RPC model is also expected to be carried out on more complex datasets, as well as on unsupervised feature selection [53] in the near future.

Acknowledgments

The authors appreciate very much the advice from the machine learning team in NLPR. This work is supported in part by NSFC (No. 61273186) for C.-G. Li and B.-G. Hu, and NSFC (No. 61271430 and 61332017) for X. Mei.

References

[1] O. Dekel, C. D. Manning, and Y. Singer, “Log-linear models for label ranking,” in Proc. Adv. Neural Inf. Process. Syst., 2003, pp. 497–504.
[2] F. Lin and W. W. Cohen, “The multiRank bootstrap algorithm: Semi-supervised political blog classification and ranking using semi-supervised link classification,” Language Technol. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-LTI-08-003, 2008.
[3] H. Li, “A short introduction to learning to rank,” IEICE Trans. Inf. Syst., vol. E94-D, no. 10, pp. 1–9, 2011.
[4] K. Harvelin and J. Kekalainen, “IR evaluation methods for retrieving highly relevant documents,” in Proc. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2000, pp. 41–48.
[5] R. Herbrich, R. Graepel, and K. Obermayer, “Large margin rank boundaries for ordinal regression,” in Advances in Large Margin Classifiers. Cambridge, MA, USA: MIT Press, 2000.
[6] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, “Learning to rank using gradient descent,” in Proc. 22nd Int. Conf. Mach. Learn., 2005, pp. 89–96.
[7] Z. Cao, T. Qin, T. Y. Liu, M. F. Tsai, and H. Li, “Learning to rank: From pairwise approach to listwise approach,” in Proc. 24th Int. Conf. Mach. Learn., 2007, pp. 129–136.
[8] F. Xia, T. Y. Liu, J. Wang, W. Zhang, and H. Li, “Listwise approach to learning to rank: Theory and algorithm,” in Proc. 25th Int. Conf. Mach. Learn., 2008, pp. 1192–1199.
[9] S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine,” Comput. Netw., vol. 30, nos. 1–7, pp. 107–117, 1998.
[10] B.-G. Hu, H. B. Qi, Y. Wang, and S. H. Yang, “A generalized-constraint neural network model: Associating partially known relationships for nonlinear regression,” Inf. Sci., vol. 179, pp. 1929–1943, 2009.
[11] W. Kotlowski and R. Slowiński, “On nonparametric ordinal classification with monotonicity constraints,” IEEE Trans. Knowl. Data Eng., vol. 25, no. 11, pp. 2576–2589, Nov. 2013.
[12] A. Y. Zinkevych and A. N. Gorban. (2010). Nonlinear quality of life index [EB/OL], New York [Online]. Available: http://arxiv.org/abs/1008.4063
[13] Microsoft Research [Online]. Available: http://research.microsoft.com/en-us/um/beijing/projects/letor/
[14] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf, “Ranking on data manifolds,” in Proc. Adv. Neural Inf. Process. Syst. 16, 2004, pp. 169–176.
[15] F. Lin, C. Chen, D. Cai, X. He, W. Liu, and J. Luo, “Efficient manifold ranking for image retrieval,” in Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2011, pp. 525–534.
[16] X. Q. Cheng, P. Du, J. F. Guo, X. F. Zhu, and Y. X. Chen, “Ranking on data manifold with sink points,” IEEE Trans. Knowl. Data Eng., vol. 25, no. 1, pp. 177–191, Jan. 2013.
[17] C. Bishop, Pattern Recognition and Machine Learning. New York, NY, USA: Springer, 2006.
[18] R. Baeza-Yates and B. Ribeiro-Neto, Information Retrieval. Reading, MA, USA: Addison-Wesley, 1999.
[19] S. Lin, “Rank aggregating methods,” Wiley Interdisciplinary Rev.: Comput. Statist., vol. 2, no. 5, pp. 555–570, 2010.
[20] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” J. Mach. Learn. Res., vol. 3, pp. 1157–1182, 2003.
[21] M. N. Volkovs and R. S. Zemel, “New learning methods for supervised and unsupervised preference aggregation,” J. Mach. Learn. Res., vol. 15, pp. 1135–1176, 2014.
[22] C. Dwork, R. Kumar, M. Naor, and S. Sivakumar, “Rank aggregation methods for the web,” in Proc. 10th Int. Conf. World Wide Web, 2001, pp. 613–622.
[23] A. Klementiev, D. Roth, and K. Small, “Unsupervised rank aggregation with distance-based models,” in Proc. 25th Int. Conf. Mach. Learn., 2008, pp. 472–479.
[24] R. Unnikrishnan, C. Pantofaru, and M. Hebert, “Toward objective evaluation of image segmentation algorithms,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 6, pp. 929–944, Jun. 2007.
[25] W. N. Venables and B. D. Ripley, Modern Applied Statistics with S, 4th ed. New York, NY, USA: Springer, 2002.
[26] N. Hurst and W. Harman, “Principal curves,” J. Amer. Stat. Assoc., vol. 84, no. 406, pp. 502–516, 1989.
[27] B. Kegl, A. Krzyżak, T. Linder, and K. Zeger, “Learning and design of principal curves,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 3, pp. 281–297, Mar. 2000.
[28] A. N. Gorban and A. Y. Zinkevych, “Principal graphs and manifolds,” in Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques, vol. I, E. S. Olivas, J. D. M. Guerrero, M. M. Sober, J. R. M. Benedito, and A. J. S. López, Eds. New York, NY, USA: Inf. Sci. Ref., 2010, pp. 28–59.
[29] J. D. Banfield and A. E. Raftery, “Ice floe identification in satellite images using mathematical morphology and clustering about principal curves,” J. Amer. Stat. Assoc., vol. 87, no. 417, pp. 7–16, 1992.
[30] P. Delicado, “Another look at principal curves and surfaces,” J. Multivariate Anal., vol. 77, no. 1, pp. 84–116, 2001.
[31] J. Einbeck, G. Tutz, and L. Evers, “Local principal curves,” Statist. Comput., vol. 15, no. 4, pp. 301–313, 2005.
[32] R. Tibshirani, “Principal curves revisited,” Statist. Comput., vol. 2, no. 4, pp. 183–190, 1992.
[33] P. Gould, “Letting the data speak for themselves,” Ann. Assoc. Amer. Geographers, vol. 71, no. 2, pp. 166–176, 1981.
[34] B.-G. Hu, X. Che, H. Zhang, D. Guo, and D. Yu, “Rank entropy-based decision trees for monotonic classification,” IEEE Trans. Knowl. Data Eng., vol. 24, no. 11, pp. 2052–2064, Nov. 2012.
[35] B.-G. Hu, G. K. I. Mann, and R. G. Gosine, “Control curve design for nonlinear (or fuzzy) proportional actions using spline-based functions,” Automatica, vol. 34, no. 9, pp. 1125–1133, 1998.
H. Daniels and M. Velikova, “Monotone and partially monotone neural networks,” IEEE Trans. Neural Netw., vol. 21, no. 6, pp. 906–917, Jun. 2010.

Y.-J. Qu and B.-G. Hu, “Generalized constraint neural network regression model subject to linear priors,” IEEE Trans. Neural Netw., vol. 22, no. 12, pp. 2447–2459, Dec. 2011.

L. F. Richardson, “The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam,” Philos. Trans. Roy. Soc. London Ser. A, vol. 210, pp. 307–357, 1910.

T. A. Pastva, “Bézier curve fitting,” M.S. thesis, Naval Postgraduate School, Monterey, CA, USA, 1998.

S. Boyd and L. Vandenberghe, Convex Optimization. New York, NY, USA: Cambridge Univ. Press, 2004.

H. A. Priestley, “Chapter 2: Ordered sets and complete Lattices-A primer for computer science,” in Algebraic and Coalgebraic Methods in the Mathematics of Program Construction, R. Backhouse, R. Crole, J. Gibbons, Eds. New York, NY, USA: Springer, 2002, pp. 21–78.

P. M. Fitzpatrick, Advanced Calculus. Pacific Grove, CA, USA: Brooks/Cole, 2006.

A. Cambibi, D. T. Luc, and L. Martein, “Order-preserving transformations and applications,” J. Optim. Theory Appl., vol. 118, no. 2, pp. 275–293, 2003.

T. W. Anderson, An Introduction to Multivariate Statistical Analysis. Hoboken, NJ, USA: Wiley, 2003.

G. Farin, Curves and Surfaces for Computer Aided Geometric Design. 4th ed. San Diego, CA, USA: Academic Press, 1997.

J. H. Lee, “Analyses of multiple evidence combination,” in Proc. 20th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 1997, pp. 267–276.

E. Zeidler, Nonlinear Functional Analysis and Its Applications. New York, NY, USA: Springer, 1988.

M. A. Jenkins and J. F. Traub, “A Three-stage algorithm for real polynomials using quadratic iteration,” SIAM J. Numer. Anal., vol. 7, no. 44, pp. 546–566, 1970.

M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, Nonlinear Programming: Theory and Algorithms. Hoboken, NJ, USA: Wiley, 2006.

G. H. Golub and C. F. van Loan, Matrix Computations, 3rd ed. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 1996.

E. Garfield, “The history and meaning of the journal impact factor,” J. Amer. Med. Assoc., vol. 293, no. 1, pp. 90–93, 2006.

C. T. Bergstrom, J. D. West, and M. A. Wiseman, “The eigenfactor metrics,” J. Neurosci., vol. 28, no. 45, pp. 11433–11434, 2008.

J. Tang and H. Liu, “An unsupervised feature selection framework for social media data,” IEEE Trans. Knowl. Data Eng., vol. 26, no. 12, pp. 2914–1927, Dec. 2014.

Chun-Guo Li received the BSc and MSc degrees in mathematics from Hebei University, Baoding, China, in 2003 and 2006, respectively. She is currently working toward the PhD degree in computer science at the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, Beijing, China. From 2008, she worked as a lecturer in Hebei University. Her research interests are in machine learning and data mining. She is a member of the IEEE.

Xing Mei received the PhD degree from the Institute of Automation, Chinese Academy of Sciences (CASIA), in 2009. During March 2009–June 2011, he was a researcher at the Samsung Advanced Institute of Technology (SAIT), China Lab. He is currently an assistant researcher at NLPR, CASIA, and a research scholar at the University at Albany, State University of New York. His research interests lie in computer graphics and computer vision.

Bao-Gang Hu received the MSc degree from the University of Science and Technology, Beijing, China, in 1983, and the PhD degree from McMaster University, Canada, in 1993, all in mechanical engineering. From 1994 to 1997, he was a research engineer and senior research engineer at C-CORE, Memorial University of Newfoundland, Canada. Currently, he is a professor with the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Science, Beijing, China. From 2000 to 2005, he was the Chinese director of LIAMA (the Chinese-French Joint Laboratory for Computer Science, Control, and Applied Mathematics). His main research interests include pattern recognition and plant growth modeling. He is a senior member of the IEEE.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.