# Direct Optimization of Evaluation Measures in Learning to Rank using Particle Swarm

Óscar Alejo[1], **Juan M. Fernández-Luna[2]**,
Juan F. Huete[2], Ramiro Pérez-Vázquez[3]

[1] Universidad de Cienfuegos, Cuba
[2] Universidad de Granada, Spain
[3] Universidad Central de las Villas, Cuba

TIR'10 Workshop – Bilbao, August 31st, 2010

# In this presentation…

RankPSO:

- A new Learning to Rank (L2R) method based on
- Particle Swarm Optimization (PSO).

- Method, experimentation and comparisons.

- Main contribution: application of PSO to L2R with similar results to those found in the specilized literature.

# Layout

I. Learning to Rank.
II. Particle Swarm Optimization.
III. Description of RankPSO approach.
IV. Experimentation and evaluation.
V. Conclusions and future works.

# Layout

I. **Learning to Rank.**
II. Particle Swarm Optimization.
III. Description of RankPSO approach.
IV. Experimentation and evaluation.
V. Conclusions and future works.

# Learning to Rank (L2R)

- Ranking is a central problem in many IR applications:
  - document retrieval,
  - collaborative filtering,
  - key term extraction,
  - definition finding,
  - important email routing,
  - sentiment analysis,
  - product rating,
  - anti web spam, …

# Learning to Rank (L2R)

Ranking problem in Document Retrieval:

> Defining a representative order among documents, taking into account relevant degree between each document and the user's query, obtaining the retrieval list, in which the relevant documents are in the highest positions with regard to less relevant document or irrelevant at all.

# Learning to Rank (L2R)

Learning to Rank:

> The application of supervised learning methods to automatically learn an effective ranking model based on training data and then apply it to test data.

# Learning to Rank (L2R)

- In L2R, the input is
  - A set of queries,
  - list of retrieved documents, and
  - relevant judgments (set of labels) for each document in each query.
- The objective:
  - To find a ranking model (function) that is able to optimize some IR evaluation measures.
  - This ranking function is applied to test data.

# Learning to Rank (L2R)

- Classification of L2R methods:
  - Pointwise Approach, which transforms ranking to classification or regression on single documents;
  - Pairwise Approach, which formalizes ranking as classification on document pairs;
  - Listwise Approach, which directly minimizes a loss function defined on document lists.

# Learning to Rank (L2R)

- Listwise Approach:
  - Probabilistic models for ranking.
  - Direct optimization of evaluation measures.
    - Minimization of loss functions upper bounding, considering loss functions defined on IR measures
    - Approximation of IR measures by means of an easy-to-handle function.
    - Specially designed technologies for optimizing non-smooth IR measures:
      - Smoothing approaches.
      - Smoothing approaches using Genetic Programming.
      - Smoothing approaches for descending gradient.

# Layout

I. Learning to Rank.

**II. Particle Swarm Optimization.**

III. Description of RankPSO approach.

IV. Experimentation and evaluation.

V. Conclusions and future works.

# Particle Swarm Optimization

- Conventional computing is sometime not capable of solving real world problems because
  - They present incomplete or noisy data, and
  - They are multi-dimensional problems.

# Particle Swarm Optimization

- Natural computing seems to be the replacements of such classical techniques in solving these problems.

- ***Basically, simple elements that can solve difficult problems of the real world working together.***

# Particle Swarm Optimization

- There are some types of them:
  - Epigenesis: we would like an intricate structure and to do so we perform a tentative learning. Artificial Neural Networks.
  - Phylogeny: competition of agents on survival of the fittest. Evolutionary algorithms.
  - Ontogeny: the adaptation of a special organism to its environment is happened. Genetic algorithms and Particle Swarm Optimization.

# Particle Swarm Optimization

- In general, there are also some drawbacks:
  - There is no guarantee in finding an optimal solution.
  - High computational costs.

# Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique developed by Russell C. Eberhart and James Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling.

# Particle Swarm Optimization

The optimization problems are represented by a group of individuals searching for food, for example.

A candidate solution is presented as a particle.

PSO uses a collection of flying particles (changing solutions) in a search area (current and possible solutions) as well as the movement towards a promising area in order to get to a global optimum.

# Particle Swarm Optimization

Particles communicate their best solution, and the members of the group follow a combination of the group's previous best and their own previous best, with an additional stochastic element to assist exploration.

# Particle Swarm Optimization

- A **particle**, i, from a swarm **δ** is composed of:
  - A **position vector, $x_i$** (coordinates in the search space),
  - a **vector of velocity, $v_i$** (displacement of that position),
  - a **memory of the best solution, $p_i$,** found by **particle i**, and
  - a **memory of the best solution, $g_{best}$,** found by the **swarm**.

# Particle Swarm Optimization

- PSO Algorithm:

1. Create a population of particles uniformly distributed in the search space.
2. Evaluate each particle's position according to the objective function.
3. If a particle's current position is better than its previous best position, update it.
4. Determine the best particle (according to the particle's previous best positions).
5. Update particles' velocities.
6. Move particles to their new positions.
7. Go to step 2 until stopping criteria are satisfied.

# Particle Swarm Optimization

Particle's position updating:

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)}$$

Particle's velocity computation:

$$v_i^{(t+1)} = v_i^{(t)} + c_1 n_1 (p_i - x_i^{(t)}) + c_2 n_2 (g_{best} - x_i^{(t)})$$

Inertia | Personal Influence | Social Influence

- $c_1$ and $c_2$, coefficient of acceleration, determining the balance between the influence of the individual's knowledge ($c_1$) and that of the group ($c_2$).
- $n_1$ and $n_2$, uniformly random numbers.

# Layout

I. Learning to Rank.
II. Particle Swarm Optimization.
**III. Description of RankPSO.**
IV. Experimentation and evaluation.
V. Conclusions and future works.

# Description of RankPSO

- Notation:
  - $\boldsymbol{Y} = \{r_1, r_2, \ldots, r_k\}$ the set of ranks, where $k$ denotes the # of ranks.
  - Total order between the them, i.e. $r_k > r_{k-1} > \ldots > r_1$ .
  - $\boldsymbol{Q} = \{q_1, q_2, \ldots, q_m\}$ is the set of queries in the training set.
  - $q_i$ = list of terms $\{t_1, t_2, \ldots, t_{h(qi)}\}$ ($h(q_i)$ is the #of terms in the $i^{th}$ query).
  - $q_i$ is associated to a list of retrieved documents $\mathbf{d}_i = \{d_{i1}, d_{i2}, \ldots, d_{in(qi)}\}$ and a list of labels $\mathbf{y}_i = \{y_{i1}, y_{i2}, \ldots, y_{in(qi)}\}$, where $n(q_i)$ denotes the sizes of lists $\mathbf{d}_i$ and $\mathbf{y}_i$, $\mathbf{d}_i \subseteq \boldsymbol{D}$.
  - $\boldsymbol{D}$ is the set of all rankings for all the queries in $\boldsymbol{Q}$.

# Description of RankPSO

- Notation:
  - $d_{ij} \in \mathbf{d}_i$ denotes the $j^{th}$ document in $\mathbf{d}_i$.
  - $y_{ij} \in \mathbf{y}_i$ is the label of document $d_{ij}$.
  - A feature vector $\phi(q_i,d_{ij})$ is created from each query-document pair $(q_i, d_{ij})$, $i$=1, 2,..., m; $j$=1, 2,..., $n(q_i)$.
  - The training set is noted as S = $(q_i, d_i, y_i)$, i= 1,...,m.

# Description of RankPSO

- Aim:
  - $\pi_i$ is the prediction made by the ranking model on $\mathbf{d}_i$ for $q_i$.
  - $\Pi_i$ = set of all possible predictions on $\mathbf{d}_i$.
  - $\pi_i(j)$ to denote the position of item $j$ (i.e. $d_{ij}$).
  - Objective: obtaining a prediction $\pi_i \in \Pi_i$ for $q_i$ and $\mathbf{d}_i$ using the ranking model.
  - The ranking model: $f(q_i, d_{ij}) = w^T \phi(q_i, d_{ij})$.

    (Linear combination of the features)

# Description of RankPSO

- Objective:
  - In the ranking of query $q_i$,
    1. we assign a score to each of the documents using $f(q_i,d_{ij})$ and
    2. sort out the documents based on their scores.
    3. We then obtain a prediction $\pi_i$.

# Description of RankPSO

- Goodness of the Ranking Model:
  - Most common IR Evaluation Measures: Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG) and Precision at n (P@n).
  - Given a ranking from the ranking model and for a query, the evaluation measures are computed.
  - $E(\pi_i, \mathbf{y}_i) \in [0,1]$ is used to represent the evaluation measures.
  - $E$ measures the agreement between $\pi_i$ and $\mathbf{y}_i$ (ground truth).

# Description of RankPSO

1: Input: , $E$ and $T$
2: **for each** particle $i$ **do**
3: Randomly initialize $v_i$, $x_i = p_i$
4: Update $g_{best}$
5: **end for each**
6: **for** t = 1,...,$T$
7: **for each** particle $i$ **do**
8: Update $i$ with $x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)}$ and $v_i^{(t+1)} = w\, v_i^{(t)} + c_1\, n_1\, o(p_i - x_i^{(t)}) + c_2\, n_2\, o(g_{best} - x_i^{(t)})$
9: Evaluate $x_i$ on $S$ *with the fitness function* $R(f)$.
10: Update $p_i$
11: Update $g_{best}$
12: **end for each**
13: **end for**
14: Build the ranking model $f$ with the position vector $g_{best}$
15: Output: $f$

# Experimentation and Evaluation

- Test collection: OSUMED, MEDLINE subset.
- It belongs to LETOR dataset. Standard in the L2R evaluation.
- There are 106 queries in the collection.
- For each query, there are a number of associated documents.
- The relevance degrees of documents with respect to the queries are judged by humans, on three levels: *definitely relevant*, *partially relevant*, or *not relevant*.
- There are 16,140 query-document pairs with relevance labels, and 45 extracted features.
- We extracted 4 features from each query-document pair, also standard in the literature.

# Experimentation and Evaluation

| Feature ID | Description |
|---|---|
| 2 | $\sum_{q_i \in q \cap d} \log(c(q_i, d) + 1)\ in\acute{}title\acute{}$ |
| 4 | $\sum_{q_i \in q \cap d} \log(\frac{c(q_i, d)}{\lvert d \rvert} + 1)\ in\acute{}title\acute{}$ |
| 8 | $\sum_{q_i \in q \cap d} \log(\frac{c(q_i, d)}{\lvert d \rvert} \cdot \log(\frac{\lvert C \rvert}{df(q_i)}) + 1)\ in\acute{}title\acute{}$ |
| 3 | LMIR with DIR smoothing 'title + abstract' |

c(qi,d) being the frequency of the query term $q_i$ in document d, C the collection, df(·) the frequency of a term in a document, and |·| the cardinality of the corresponding set.

# Experimentation and Evaluation

- Evaluation Measures:
  - Precision at position n (P@n):

$$P@n = \frac{\#relevant\ docs\ in\ top\ n\ results}{n}.$$

  - Mean Average Precision (MAP):

$$AP = \frac{\sum_{n=1}^{N}(P@n * rel(n))}{\#total\ relevant\ docs\ \ for\ this\ query}$$

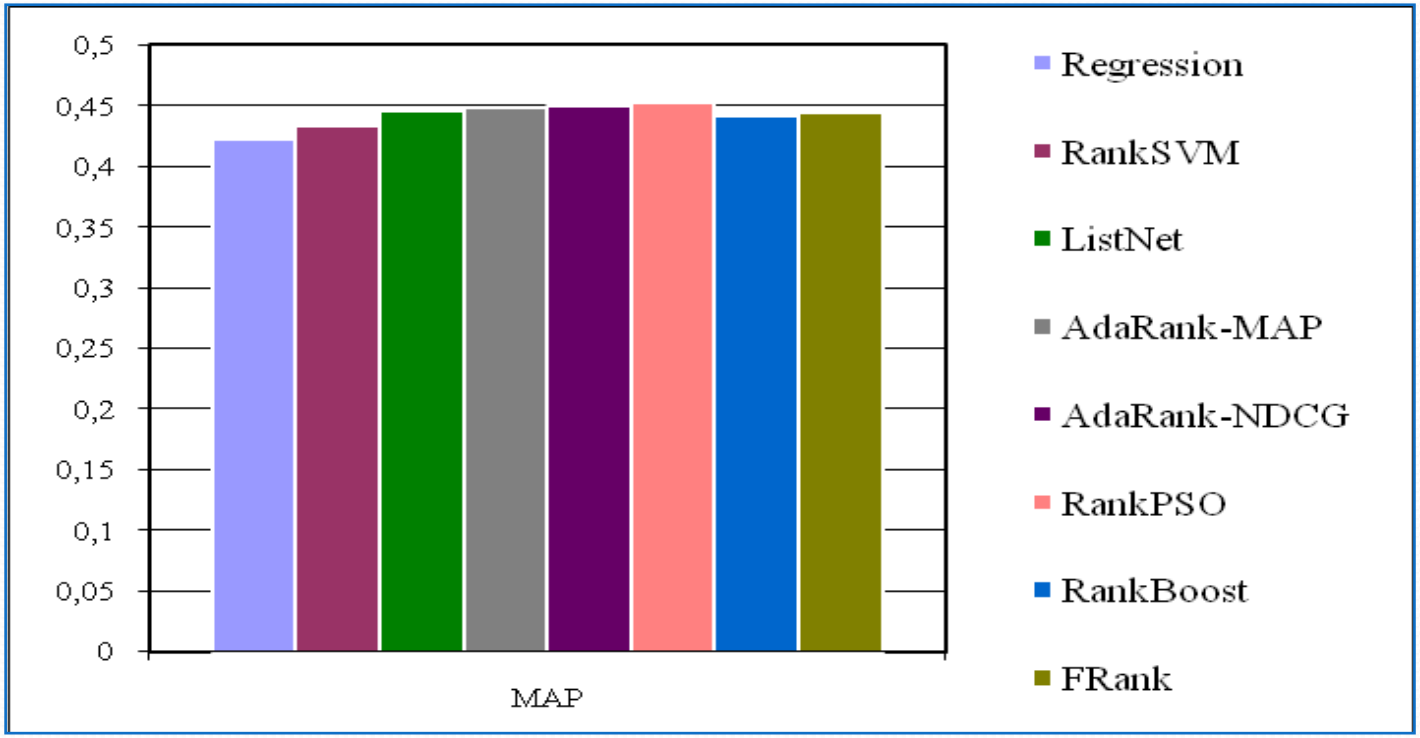
  - Normalized Discounted Cumulative Gain (NDCG):

$$NDGC(n) \equiv Z_n \sum_{j=1}^{n} \frac{2^{r(j)}-1}{\log(1+j)}$$

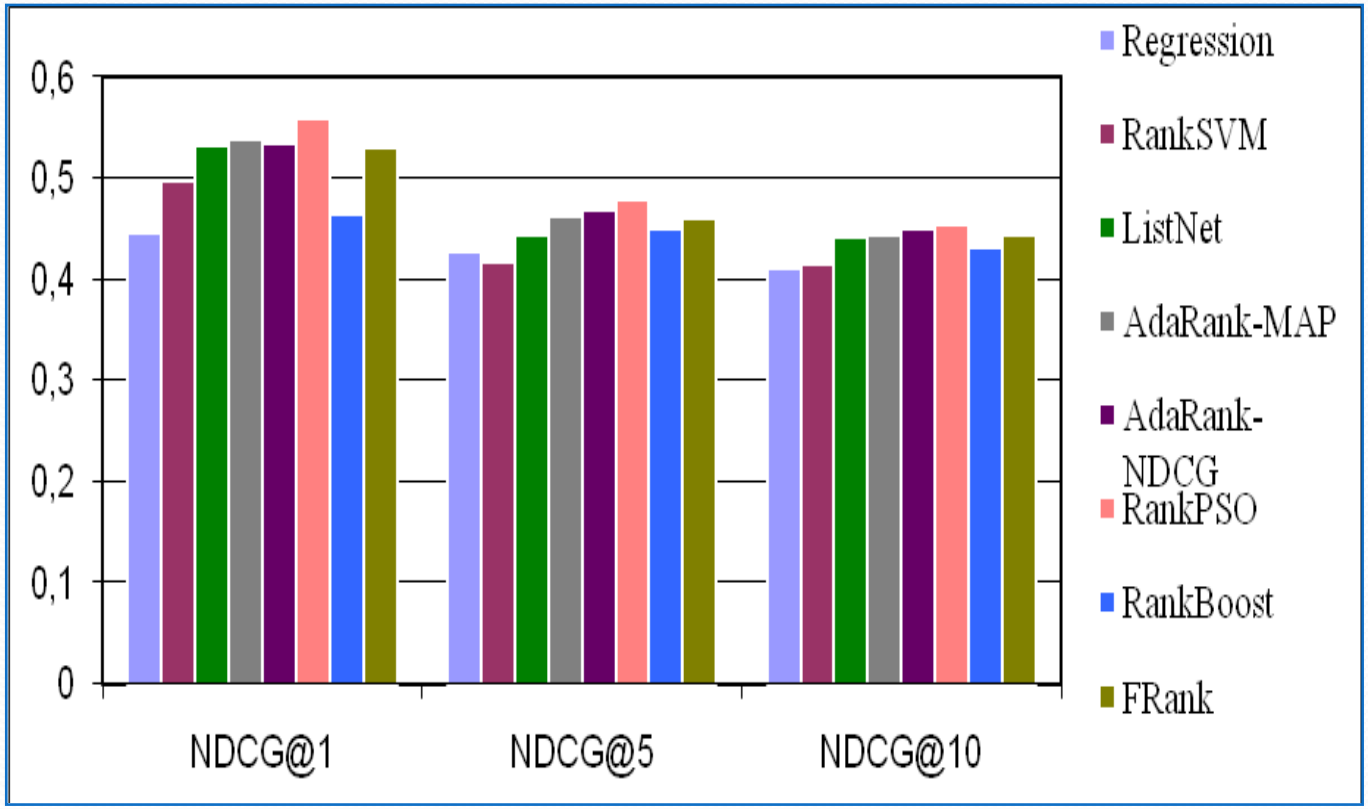
# Experimentation and Evaluation

- For the learning or training process, 5-fold cross-validation, experiments were performed. These prefixed 5-folds in OHSUMED were taken from the version "*QueryLevelNorm*".
- Comparison with those algorithms that have got their assigned scores for each ranking function applied to each query-document published in the LETOR website:
  - Pointwise approach: Regression;
  - Pairwise approaches: RankSVM, RankBoost, FRank;
  - Listwise approaches: ListNet, with loss minimization, and AdaRank, with direct optimization of IR measures.

# Experimentation and Evaluation



# Experimentation and Evaluation



# Experimentation and Evaluation

| Algorithms | P@1 | P@2 | P@3 | P@4 | P@5 | P@6 | P@7 | P@8 | P@9 | P@10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Regression | 0.597 | 0.601 | 0.577 | 0.561 | 0.534 | 0.505 | 0.500 | 0.484 | 0.475 | 0.467 |
| RankSVM | 0.597 | 0.549 | 0.543 | 0.544 | 0.532 | 0.525 | 0.510 | 0.493 | 0.492 | 0.486 |
| ListNet | 0.652 | 0.609 | **0.602** | 0.575 | 0.550 | 0.537 | 0.527 | 0.524 | 0.514 | 0.498 |
| AdaRank-MAP | 0.634 | 0.596 | 0.590 | 0.589 | 0.567 | 0.557 | 0.539 | 0.524 | 0.508 | 0.498 |
| AdaRank-NDCG | **0.672** | **0.624** | 0.598 | 0.584 | 0.577 | 0.556 | **0.551** | **0.535** | **0.521** | **0.509** |
| RankBoost | 0.558 | 0.548 | 0.561 | 0.558 | 0.545 | 0.530 | 0.524 | 0.513 | 0.502 | 0.497 |
| FRank | 0.643 | 0.620 | 0.593 | 0.584 | 0.564 | 0.552 | 0.545 | 0.525 | 0.515 | 0.502 |
| RankPSO | **0.672** | 0.619 | 0.593 | **0.593** | **0.579** | **0.558** | 0.547 | 0.533 | **0.521** | 0.506 |

# Experimentation and Evaluation

- Statistic tests were applied to determine the significance in the precision at query level.
- Wilcoxon test, even for *k* related samples, using Friedman test, considering MAP as evaluation measure because it represents the mean average precision of the full ranking for each one of the queries, and not only for the first positions.
- In the tests analysis, the statistic significance was considered with p-value<0.05.
- RankPSO has **significant improvement** in terms of precision compared to RankSVM, RankBoost and Regression methods;
- nevertheless, it do **not have significant differences** with AdaRank-MAP, AdaRank-NDCG, ListNet y FRank.

# Conclusions and Future Works

In this paper, we have proposed a new method called **RankPSO** for **Learning to Rank**.

This approach is based on **Particle Swarm Optimization** and allows direct optimization of evaluation measures used in IR.

# Conclusions and Future Works

- Considering empirical results, we could conclude that **RankPSO is just as good as the similar direct optimization methods**.
- The main advantages are **easy implementation**, it allows a **direct optimization of any performance measure** and the **ranking model builds a linear function**.
- Methods based on direct optimization of evaluation measures can always outperform conventional methods.
- However, no significant difference exists among the performances of the direct optimization methods themselves. Perhaps ceiling effect??

# Conclusions and Future Works

- As future works:
  - Comparison with other state-of-the-art methods.
  - More experiments with medium and large scale datasets, to further verify the performance of RankPSO.
  - Proposal of new L2R models based on Multi-objective PSO.
  - Searching for the application of new bio-inspired algorithms at L2R for IR.
  - To conceive new ranking models taking into account not only the queries, the associated list of documents for these queries and relevant judgments, but also the context where the queries are formulated.

# That's all

Thank you for your attention

Questions?