Universidad de Cienfuegos "Carlos Rafael Rodríguez"
Facultad de Ingeniería
Carrera de Ingeniería Informática

"Desarrollo de un nuevo método de Aprendizaje de la Ordenación, basado en el algoritmo de optimización global Supernova."

Trabajo de diploma para optar por el título de Ingeniería en Informática

Autora:
Diana Lourdes Pernús Alonso

Tutor:
Msc. Oscar José Alejo Machado

Cienfuegos, Cuba
Curso 2011-2012

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo al Departamento de Informática de la Facultad de Ingeniería en la Universidad de Cienfuegos “Carlos Rafael Rodríguez”, para que hagan el uso que estimen pertinente con el trabajo de diploma.

Para que así conste firmo la presente a los 4 días del mes de junio del 2012.

Diana Lourdes Pernús Alonso

Nombre completo del autor

Msc. Oscar José Alejo Machado

Nombre completo del tutor

Los abajo firmantes certificamos que el presente trabajo ha sido revisado según acuerdo de la dirección de nuestro centro y el mismo cumple los requisitos que debe tener un trabajo de esta envergadura referente a la temática señalada.

Firma Tutor

Firma ICT

Firma Vicedecano

"*No creo haber conseguido ya la meta*
ni me considero perfecto,
más bien sigo adelante
con la esperanza de alcanzarlo"

(Filipenses 3, 12)

Agradecimientos

A todas las personas que han contribuido a mi formación como persona y profesional, especialmente a mi familia.

A los que han sido compañeros de camino dándome alegría, apoyo, consejos, esperanza…

A mi tutor Oscarito que me motivó y guió durante la realización de este trabajo.

Dedicatoria

A Dios que cada día me hace feliz con detalles de Su amor. A mi familia que es Su mayor regalo.

Resumen

Uno de los principales problemas en el Aprendizaje de la Ordenación para la Recuperación de Información (R.I.) es desarrollar algoritmos que construyan modelos de ordenación. Con el objetivo de abordar esta tarea, la comunidad científica trabaja en tres grandes vertientes; la mayoría de los métodos referenciados dentro de las categorías de aproximación por puntos (*pointwise approach*), aproximación por pares (*pairwise approach*) y aproximación por listas minimizando la función de pérdida (*listwise loss minimization*) son capaces solamente de entrenar modelos de ordenación minimizando una función de pérdida relativa en cierto grado a las medidas de desempeño, por lo que en la actualidad se trabaja fundamentalmente en la optimización directa de medidas de desempeño utilizadas en la R.I.

Los métodos desarrollados presentan dificultades para alcanzar una optimización global y no muestran estabilidad en cuanto a precisión en sus resultados para los diferentes conjuntos de datos, por lo que en el presente trabajo se propone un nuevo método de Aprendizaje de la Ordenación basado en Supernova, un novedoso algoritmo de optimización global. Este método se denomina RankSupernova y permite optimizar directamente cualquier medida de desempeño utilizada en la R.I.

El método propuesto muestra una buena estabilidad en sus resultados al ser evaluado su desempeño en los conjuntos de datos de LETOR 3.0. Se realizó un análisis estadístico donde se obtuvo que RankSupernova supera significativa e indistintamente a los métodos Regression, RankSVM y AdaRankNDCG; además, no es superado significativamente por ninguno de los métodos de aprendizaje considerados y el valor de su media (sustentado en la precisión a nivel de consulta) se encuentra siempre en las tres mejores posiciones de la ordenación.

Índice de contenido

Índice de tablas

Índice de figuras

Introducción

En el pasado siglo la humanidad comenzó a vivir la era de la información. Cada día el volumen de información concerniente a un tema determinado aumenta considerablemente por lo que se hace indispensable su obtención, procesamiento y distribución no sólo de forma rápida, sino también eficiente. En este contexto, las aplicaciones de Recuperación de Información (R.I.) enfrentan diversos problemas, la ordenación constituye uno de los más significativos.

En la recuperación de documentos, este problema consiste en definir un orden representativo entre ellos, teniendo en cuenta su relevancia con respecto a la consulta de un usuario, de forma tal que la información relevante sea colocada en mejor posición respecto a aquella que es menos relevante o irrelevante.

Ante esta problemática, un grupo de investigadores comenzaron a desarrollar diversos modelos de R.I. entre los años 1970s y 1990s. Estos modelos se agruparon en dos grupos atendiendo a su concepción y a la relación que tenían con las consultas de los usuarios: modelos dependientes de la consulta y modelos independientes de la consulta.

Dentro de los modelos dependientes de la consulta están: los modelos booleanos, los vectoriales, los probabilísticos, los basados en el lenguaje, entre otros; y como modelos independientes de la consulta se pueden mencionar: *PageRank* y *BrowseRank*.

En la mayoría de estos modelos de R.I. un documento es representado como un conjunto de palabras claves representativas (términos que instancian rasgos o características de los documentos) por las cuales es indexado, y se define una función de ordenación (o función de recuperación) que permite asociar un grado de relevancia con un documento y una consulta.

Con el paso del tiempo, los modelos de ordenación se complejizan: se consideran nuevos rasgos para su formulación, se hace necesario hacerlos variables y que además aprendiesen a partir de datos de entrenamiento. Todo esto unido a la incorporación de juicios de relevancia a las colecciones documentales en 1994, hace que se comience a pensar en emplear técnicas de aprendizaje automático.

Los métodos que se comenzaron a desarrollar desde entonces, utilizan técnicas de aprendizaje automático y dieron origen a un nuevo tópico vigente de investigación dentro de la rama de la Inteligencia Artificial y la Recuperación de Información: el Aprendizaje de la Ordenación, denominado en inglés *Learning to Rank*.

Estos nuevos métodos de Aprendizaje de la Ordenación se basan en el aprendizaje supervisado y persiguen sobre cualquier escenario, crear de forma automática, un modelo de ordenación usando datos de entrenamiento, técnicas de aprendizaje automático y medidas de desempeño de la R.I. como: MAP (*Mean Average Precision*), NDCG (*Normalized Discounted Cumulative Gain*) y P@n (*Precision at n*).

El Aprendizaje de la Ordenación se enmarca inicialmente en tres categorías principales: Aproximación por Puntos (*Pointwise Approach)*, que trabaja la ordenación sobre documentos simples; Aproximación por Pares (*Pairwise Approach)*, en la cual se transforma o formaliza el problema en una clasificación binaria sobre pares de documentos y Aproximación por Listas (*Listwise Approach)*, en la cual se minimiza directamente una función de pérdida definida sobre una lista de documentos.

La mayoría de los métodos referenciados dentro de estas categorías son capaces solamente de entrenar modelos de ordenación minimizando una función de pérdida relativa en cierto grado a las medidas de desempeño. Sin embargo, idealmente un algoritmo de aprendizaje debiese entrenar un modelo de ordenación que directamente optimizara medidas de desempeño con respecto a los datos de entrenamiento.

A partir del año 2005 se trabaja mayormente en la categoría *Listwise Approach* específicamente en la optimización directa de medidas de desempeño y se identifican tres vertientes fundamentales: (1) minimizar una función de pérdida que acote superiormente una función de pérdida básica definida sobre las medidas de R.I.; (2) aproximar las medidas de R.I. mediante funciones fáciles de manipular y (3) usar tecnologías especialmente diseñadas para optimizar medidas de R.I. no suaves. En esta última se enmarca el método que se propone en el presente

trabajo.

Aunque los métodos que optimizan directamente medidas de desempeño, han propiciado un giro contundente en el estado del arte del Aprendizaje de la Ordenación, todavía las soluciones muestran dificultad para alcanzar una aproximación global debido a que en diversas ocasiones la función objetivo es discontinua, estacionaria y presenta problemas de convexidad, por lo que el gradiente se indefine y los métodos sólo garantizan una aproximación local. A esto se une el problema archiconocido de la variabilidad de la estabilidad, en cuanto a precisión en la ordenación, de los métodos referenciados para las diferentes colecciones de datos existentes.

Es el análisis de estos antecedentes lo que lleva a plantear el siguiente **problema**: ¿Cómo optimizar directamente cualquier medida de desempeño utilizada en la R.I., logrando una estabilidad de precisión aceptable frente a los métodos de Aprendizaje de la Ordenación referenciados en la literatura?

Se define como **objeto de estudio**: el Aprendizaje de la Ordenación para la Recuperación de Información y como **campo de acción**: los métodos de Aprendizaje de la Ordenación que optimizan directamente medidas de desempeño.

Encontrar una estrategia capaz de optimizar la solución, es una tarea compleja que sugiere el uso de técnicas computacionales basadas en *SoftComputing* (Algoritmos Evolutivos, Programación Genética, Lógica difusa, entre otros).

Supernova es un novedoso algoritmo de optimización global cuya heurística está basada en el fenómeno de las supernovas y converge al mínimo global de una función no lineal de alta complejidad, por lo que se considera que dicho algoritmo puede brindarle solución a la problemática planteada y se define como **idea a defender:** la concepción de un nuevo método de Aprendizaje de la Ordenación inspirado en Supernova, será capaz de optimizar directamente cualquier medida de desempeño utilizada en la R.I. y alcanzará una estabilidad de precisión aceptable frente a los métodos referenciados en la literatura.

Partiendo de esta idea, la presente investigación se traza el siguiente **objetivo general**:

- Proponer un nuevo método de Aprendizaje de la Ordenación inspirado en Supernova, que se centre en la optimización directa de cualquier medida de desempeño utilizada en la R.I. y logre una estabilidad de precisión aceptable frente a los métodos referenciados en la literatura.

De este se derivan los siguientes **objetivos específicos:**

- Analizar el estado del arte del Aprendizaje de la Ordenación y la técnica de optimización global Supernova.
- Diseñar un nuevo método de Aprendizaje de la Ordenación tomando como base el algoritmo Supernova
- Implementar el método propuesto
- Realizar pruebas y comparaciones con colecciones de datos y métodos referenciados en la literatura

Las **tareas** a realizar para cumplir con los objetivos propuestos son:

- Estudio de las principales categorías, modelos y artículos científicos publicados en el área de investigación del Aprendizaje de la Ordenación para la R.I.
- Estudio de las principales aplicaciones y resultados del método de optimización global Supernova.
- Implementación del algoritmo Supernova adaptado al problema del Aprendizaje de la Ordenación.
- Determinación del rendimiento que se puede alcanzar con el modelo propuesto mediante su evaluación con una serie de colecciones documentales estándar.
- Comparación, utilizando métodos estadísticos, de los desempeños del modelo propuesto con otros modelos clásicos de Aprendizaje de la Ordenación, así como con los que optimizan directamente medidas de desempeño utilizadas en R.I.

El desarrollo de este método brinda la siguiente **novedad científica:** por vez primera se concibe un método de Aprendizaje de la Ordenación para la R.I., utilizando el algoritmo de optimización global Supernova.

Para el adecuado análisis y entendimiento de este documento, se ha estructurado el mismo en 3 capítulos. Los cuales hacen referencia a:

Capítulo I.- Fundamentación Teórica. En este se define la R.I., sus objetivos y los retos que enfrentan los Sistemas de R.I. Se describe la evolución que han tenido los modelos de R.I. dando lugar al Aprendizaje de la Ordenación con nuevos métodos que emplean técnicas de Inteligencia Artificial y se enmarcan en tres categorías fundamentales: aproximación por puntos, por pares y por listas. Se presenta el fenómeno físico de la supernova y la metodología empleada para el desarrollo de la investigación, así como las herramientas utilizadas. Al final se plantean algunas discusiones y trabajos futuros relacionados con las temáticas abordadas en la investigación.

Capítulo II.- Supernova en el Aprendizaje de la Ordenación. Primeramente se describe detalladamente el fenómeno físico de la supernova, lo que permite una mejor comprensión de los pasos que sigue el algoritmo de optimización basado en este fenómeno. Dicho algoritmo se adapta al problema del Aprendizaje de la Ordenación con el desarrollo del método RankSupernova, que se explica y muestra en forma de pseudocódigo al final de este capítulo.

Capítulo III.- Resultados Experimentales. Se presentan las colecciones de datos (OHSUMED, Gov) y las medidas de evaluación (MAP, P@n, NDCG) que utilizan los modelos de Aprendizaje de la Ordenación. Con el empleo de estos elementos se realizan las comparaciones del método RankSupernova con los métodos referenciados en la literatura, mostrándose el desempeño obtenido y el análisis estadístico realizado.

Finalmente se exponen las conclusiones a las que se arribó con el desarrollo de esta investigación.

Capítulo I: Fundamentación teórica

1.1 Introducción

El presente capítulo aborda de forma panorámica y conceptual, las principales investigaciones que diversos autores han venido realizando dirigidas a la Recuperación de Información (R.I.) y al Aprendizaje de la Ordenación para la R.I.

Se plantean las principales características y objetivos que persigue el Aprendizaje de la Ordenación como problema clave en la recuperación de documentos y se describen las categorías principales en las que se enmarcan los modelos que se desarrollan.

Se explica brevemente el fenómeno físico de la Supernova y se presenta la metodología empleada en el desarrollo de la investigación, así como las herramientas utilizadas. Finalmente se mencionan algunas discusiones y trabajos futuros relacionados con las temáticas abordadas en la investigación.

1.2 Recuperación de Información

La R.I. es un campo relacionado con la estructura, análisis, organización, almacenamiento, búsqueda y obtención de información [1] que se ha ido desarrollando desde la década de los sesenta del pasado siglo. Con el paso del tiempo ha sido influenciada por la popularización de Internet y los continuos y crecientes avances producidos en las tecnologías de la información que cada día almacenan y procesan mayor cantidad de información.

Su principal objetivo es proporcionarle información relevante al usuario para satisfacer su necesidad de información. El cómo lograrlo ha traído el interés de investigadores de diversas disciplinas y provocado un trabajo multi e interdisciplinar [2]. Fruto de este trabajo surgen los Sistemas de Recuperación de Información (S.R.I.) que permiten identificar aquellos documentos pertenecientes a una colección que mejor responden a las necesidades de un usuario, expresadas mediante una consulta.

La representación y organización que realizan estos sistemas deberían proveer al usuario de un fácil acceso a la información en la que se encuentre interesado.

Desafortunadamente, la caracterización de la necesidad informativa de un usuario no es un problema sencillo de resolver [3].

Los S.R.I. convencionales representan un documento como un conjunto de palabras claves representativas (términos que instancian rasgos o características de los documentos, por los cuales son indexados), y se define una función de ordenación que permite asociar un grado de relevancia al documento respecto a una consulta determinada. Esta información es utilizada para obtener un orden de los documentos colocando en mejores posiciones aquellos que contienen información más relevante.

En estos sistemas, tanto los términos de entrada como el texto de salida se pueden ordenar según ciertos criterios; el reto para los investigadores es desarrollar algoritmos que optimicen estos rasgos [4].

Para abordar este problema se han desarrollado un gran número de modelos que relacionan documentos y consultas a través de funciones de ordenación.

1.2.1 Modelos de R.I. convencionales

En el período de 1970 a 1990 surgieron varios modelos fundamentados en métodos formales. Estos se enmarcan en dos categorías:

- Modelos dependientes de la consulta:
  - Modelo Booleano (*Boolean Model*), Modelo Booleano Extendido [5]
  - Modelo del Espacio Vectorial (*Vector Space Model*) [6]
  - Modelos Probabilísticos (*Probabilistic Models*) [7]
  - Modelo BM25 [8]
  - Modelo del lenguaje estadístico (*Statistical language model*) [9]
  - *Latent Semantic Indexing* (LSI) [10]
  - Modelos del lenguaje para la RI (LMIR, *Language Models for Information Retrieval*) [11]
- Modelos independientes de la consulta
  - *PageRank* [12]
  - *TrustRank* [13]

– *BrowseRank* [14]

El modelo booleano es el más usado históricamente. Está basado en la teoría de conjuntos y el álgebra booleana. Su efectividad radica en dividir los términos de la búsqueda en conjuntos, por ello es muy fácil de implementar y entender [15]. Su problema es que devuelve resultados de todo o nada y la Recuperación de Información no es un proceso exacto.

El modelo probabilístico utiliza la teoría de probabilidades para modelar la incertidumbre del proceso de Recuperación de Información. En este, los documentos se ordenan de forma decreciente según su probabilidad de relevancia respecto a la información requerida. Sin embargo, no se usan frecuencias de términos dentro del documento ni longitud de documentos.

En el modelo de espacio vectorial, los documentos y las búsquedas se interpretan como vectores de términos, y se representa cada término en el vector con un peso *w* dentro de ese documento [15]. La función de similitud entre el documento y la consulta será el coseno del ángulo entre los vectores que los representan. La funcionalidad de este modelo radica en la elección correcta de los pesos de cada término.

Estos modelos resultaron significativos en este período de grandes retos científicos y actualmente influyen de forma directa o indirecta en la mayoría de las investigaciones que se realizan en el campo de la R.I. [16]

El hecho de que muchos de los procesos que lleva a cabo la R.I. tengan un marcado carácter impreciso e incierto (caracterización del contenido de un documento mediante vectores de términos, descripción de la necesidad de información de un usuario mediante una consulta, ...) condujo a que con el paso del tiempo, los modelos de ordenación se fueran haciendo más complejos: se consideraron nuevos rasgos para su formulación, se hizo necesario hacerlos variables y que además aprendiesen a partir de datos de entrenamiento. Todo esto unido a la incorporación de juicios de relevancia a las colecciones documentales en 1994, hizo que se comenzara a pensar en emplear técnicas de aprendizaje automático.

Los métodos que se comenzaron a desarrollar desde entonces, utilizan técnicas de aprendizaje automático y dieron origen a un nuevo tópico vigente de investigación dentro de la rama de la Inteligencia Artificial y la Recuperación de Información: el Aprendizaje de la Ordenación, denominado en inglés *Learning to Rank*.

1.3 Aprendizaje de la ordenación

1.3.1 Descripción del problema

En el proceso de aprendizaje, el sistema recibe un conjunto de consultas y los documentos recuperados correspondientes, además de las etiquetas de los documentos con respecto a las consultas. Estas etiquetas representan niveles o categorías de relevancia en un orden total.

El propósito del aprendizaje es construir un modelo de ordenación capaz de obtener los mejores resultados sobre un conjunto de datos, haciendo corresponder el conjunto ordenado según el modelo con el conjunto de documentos idealmente ordenados de acuerdo a los juicios o categorías de relevancia, que representan las preferencias de determinado usuario.

La correspondencia entre ambos conjuntos es determinada a través de una medida de desempeño, que cuantifica la relevancia del conjunto de documentos recuperados según las necesidades manifestadas por el usuario en su consulta. Entre las medidas de desempeño más utilizadas en R.I. están: MAP (*Mean Average Precision*) [17], NDCG (*Normalized Discounted Cumulative Gain*) [18] y P@n (*Precision at n*) [17]. Para una mejor comprensión de las medidas de desempeño, consulte el apartado 3.3 Medidas de Evaluación.

El objetivo del algoritmo de aprendizaje es entrenar un modelo de ordenación que directa o indirectamente optimice una de estas medidas de desempeño con respecto a los datos del conjunto de entrenamiento definido anteriormente. Específicamente, una función de ordenación de la forma $f: Q \times D \rightarrow \mathbb{R}$, asigna puntuaciones reales de adecuación a los documentos recuperados del conjunto D

con respecto a la correspondiente consulta perteneciente al conjunto Q y a continuación ordena tales documentos a partir de sus puntuaciones. [16]

El desempeño de esta función de ordenación se evalúa teniendo en cuenta el orden resultante y el orden ideal. Finalmente, el modelo realiza un ajuste general de sus parámetros, según el comportamiento evaluado. Este proceso de aprendizaje se repite hasta abarcar todos los datos de entrenamiento.

Cuando se ha ajustado el modelo de ordenación, se pasa a la etapa de prueba, en la que dada una consulta y la lista de documentos recuperados en relación a esta, el sistema debe ser capaz de retornar la lista de documentos ordenados de acuerdo a su relevancia.

Idealmente, la función de ordenación en esta etapa, alcanzará resultados suficientemente buenos en precisión que propicien la satisfacción del usuario.

Este proceso de aprendizaje puede ser comprendido mejor si se analiza el esquema de la Figura 1.1.

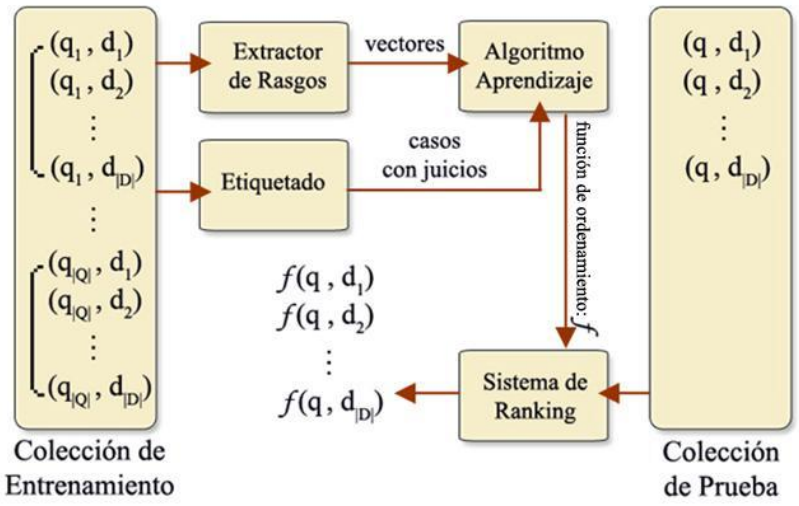


Figura 1.1 Esquema General del Aprendizaje de la Ordenación para la R.I. [16]

1.3.2 Principales categorías y modelos

Los métodos existentes de Aprendizaje de la Ordenación se dividen en tres categorías: (1) Aproximación por puntos *(Pointwise Approach)*, que realiza la ordenación sobre un par consulta - documento; (2) Aproximación por pares *(Pairwise Approach)*, en la que de manera general se transforma o formaliza el problema de la ordenación en una clasificación

binaria sobre pares de documentos y (3) Aproximación por listas *(Listwise Approach)*, en la cual se toma la lista de documentos recuperados, con respecto a una consulta, como instancias para entrenar el modelo de ordenación.

1.3.2.1 Aproximación por puntos (Pointwise approach)

En la aproximación por puntos el espacio de entrada contiene los vectores de atributos de cada par consulta - documento. El espacio de salida representa el grado de relevancia de cada documento. En el espacio de hipótesis se consideran funciones que toman los vectores de atributos y predicen un valor de relevancia del documento respecto a la consulta. La función de pérdida se obtiene a partir de problemas de regresión, clasificación o regresión ordinal. [19]

Una gran ventaja de estas aproximaciones es su sencillez, pues aplican técnicas previamente desarrolladas cuyo funcionamiento está respaldado por la comunidad de aprendizaje automático. No obstante, la sencillez constituye también su principal desventaja.

Esto es porque, conceptualmente, las tareas de regresión, clasificación y ordenación son muy distintas y se requiere de métodos más específicos que consideren relaciones de orden, pues en esto consiste el problema original. [19]

1.3.2.2 Aproximación por pares (Pairwise approach)

Una evolución de los métodos anteriores consiste en escoger pares de documentos cuya relevancia relativa con respecto a una consulta sea conocida.

En la aproximación por pares el espacio de entrada contiene pares de documentos representados por sus vectores de atributos. El espacio de salida representa los órdenes relativos de los pares de documentos con respecto a la consulta en el *ranking* esperado. Por tanto, se espera que las funciones del espacio de hipótesis sean capaces de ordenar correctamente

pares de documentos respecto a su relevancia frente a una consulta. Las funciones de pérdida, por lo general, tienden a cuantificar el número de pares mal ordenados. [19]

En el proceso de aprendizaje se toman pares de documentos de la lista recuperada y para cada par se asigna una etiqueta representando la relevancia relativa de los dos documentos [20]. Esta etiqueta puede tomar dos valores: correctamente ordenado e incorrectamente ordenado. De esta forma el problema de la ordenación queda formalizado como un problema de clasificación.

La gran ventaja de usar estos modelos es que, con frecuencia, la fuente de información única o más fiable es respecto a pares de documentos y predecir el orden relativo es más cercano a la naturaleza de la ordenación que predecir la etiqueta de la clase o el grado de relevancia. Además existen metodologías de clasificación que pueden ser directamente aplicadas [20].

Una de sus desventajas reside en el tipo de funciones de error que emplea. Estas suelen estar desconectadas de las medidas de desempeño de la R.I. o en el mejor de los casos, representan cotas superiores a las mismas.

El principal problema de estas dos primeras categorías es que los modelos desarrollados realizan la ordenación con respecto a un solo documento o a un par de estos, ignorando que el aprendizaje es una tarea de predicción sobre una lista de documentos.

1.3.2.3 Aproximación por listas (Listwise Approach)

En la aproximación por listas, el espacio de entrada contiene el grupo entero de documentos asociados con una consulta. El espacio de salida consiste en permutaciones que definen *rankings* o conjuntos de puntuaciones de relevancia para cada documento. Las funciones del espacio de hipótesis tienen como objetivo reproducir permutaciones sobre los elementos o puntuaciones de relevancia. A su vez, las funciones de

error pueden tener en cuenta la diferencia entre la permutación de predicción y la original, o bien usar métricas de evaluación en R.I. [19]

En esta categoría se presentan dos fuertes vertientes:

- *Listwise loss minimization* que minimiza una función de pérdida definida sobre las permutaciones, las cuales son diseñadas considerando las propiedades de la ordenación en la R.I.
- *Direct optimization of IR measure* en la que se trata de optimizar medidas de evaluación de R.I., o por lo menos, alguna función definida en correlación a tales medidas.

A partir del año 2005, los métodos aproximados que realizan la optimización directa de medidas de desempeño han despertado el interés de muchos investigadores y se han desarrollado importantes trabajos, que se agrupan en tres nuevas categorías dentro del Aprendizaje de la Ordenación:

- Minimizar una función de pérdida que acote superiormente una función de pérdida básica definida sobre las medidas de R.I.
- Aproximar las medidas de desempeño de R.I. mediante funciones fáciles de manipular.
- Utilizar tecnologías especialmente diseñadas para optimizar medidas de desempeño de R.I. no suaves.

Estos métodos son, a priori, los más completos y correctos con respecto al problema a solucionar.[19]

1.3.2.4 Métodos desarrollados

En la Figura 1.2, se muestra una línea de tiempo del Aprendizaje de la Ordenación, donde se reflejan los métodos desarrollados en cada una de las categorías generales a partir del año 2000 y hasta el 2009.

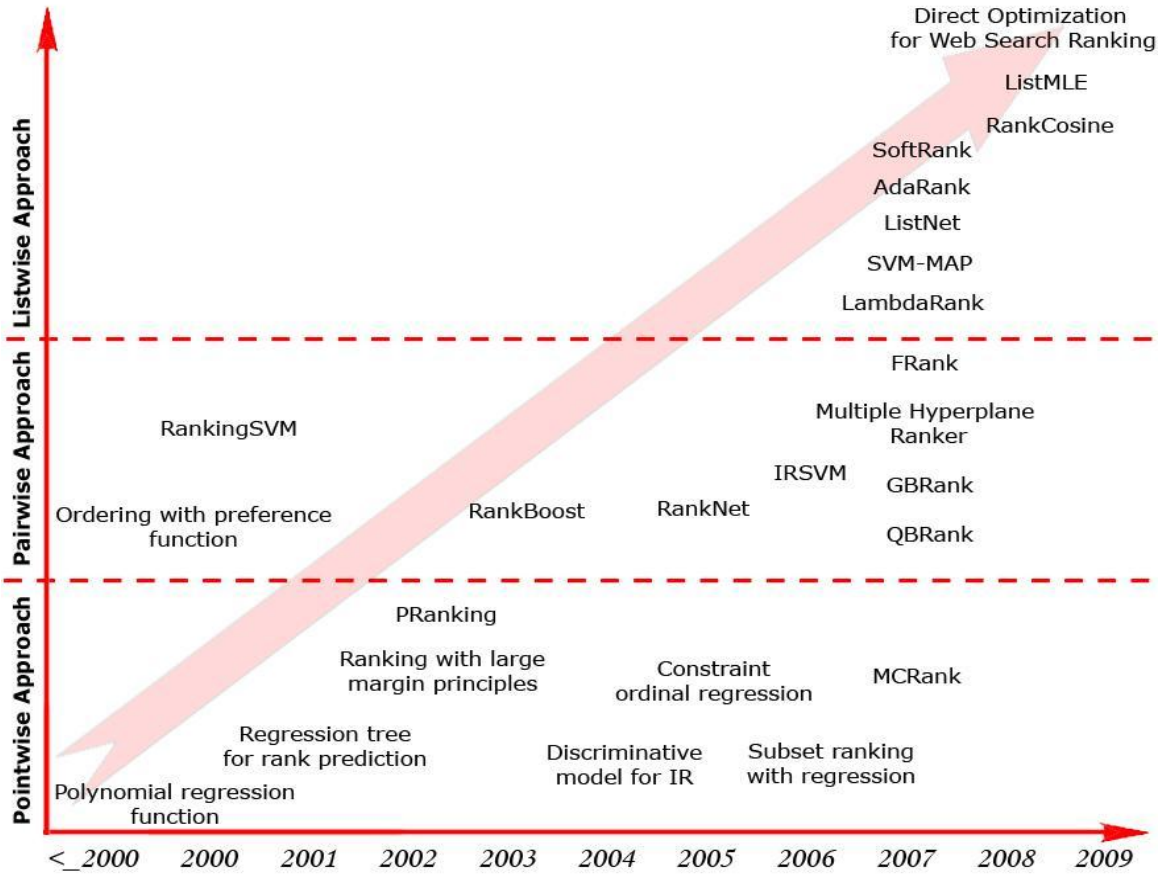


Figura 1.2: Línea de tiempo y tendencia categórica del Aprendizaje de la Ordenación. [16]

1.3.3 Problemas existentes

La mayoría de los métodos referenciados dentro de las categorías de aproximación por puntos, por pares y por listas (en la vertiente *Listwise loss minimization*) son capaces solamente de entrenar modelos de ordenación minimizando una función de pérdida relativa en cierto grado a las medidas de desempeño. Ejemplo de ello, son los métodos RankingSVM [21] y RankBoost [22] que entrenan modelos de ordenación minimizando el error de clasificación sobre pares de instancias que representan documentos.

El principal problema de esta vertiente es que la conexión entre la función de pérdida y las medidas de desempeño designadas es incierta y no necesariamente la optimización de esta función resulta en la optimización

de la medida de desempeño [23]. Además, idealmente un algoritmo de aprendizaje debiese entrenar un modelo de ordenación que directamente optimizara medidas de desempeño con respecto a los datos de entrenamiento.

Con esta perspectiva se ha venido trabajando, y actualmente la optimización directa de medidas de desempeño en el aprendizaje se ha convertido en un novedoso e interesante tópico de investigación. Varios métodos para clasificación (Ej. [24]) y ordenación (Ej. [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36]) han sido propuestos sobre la base de diferentes enfoques.

Sin embargo, a pesar de todos estos avances y logros, y aunque los métodos que optimizan directamente medidas de desempeño, han propiciado un giro contundente en el estado del arte del Aprendizaje de la Ordenación, todavía las soluciones muestran dificultad para alcanzar una aproximación global, debido a que en diversas ocasiones la función objetivo es discontinua, estacionaria y presenta problemas de convexidad, por lo que el gradiente se indefine y los métodos sólo garantizan una aproximación local.

A esto se une el problema archiconocido de la variabilidad de la estabilidad, en cuanto a precisión en la ordenación, de los métodos referenciados para las diferentes colecciones de datos existentes.

Por lo cual, el camino del Aprendizaje de la Ordenación resulta en un tópico vigente de investigación para la comunidad científica.

1.4 Supernova. Método de optimización global

Las metaheurísticas, generalmente, se inspiran en procesos del entorno. La analogía de un proceso que se ordene, o mejore su desempeño puede proveer información valiosa que permita resolver problemas de optimización de gran complejidad [37]. El fenómeno de la Supernova es un proceso que desde su inicio caótico llega a un equilibrio más ordenado. A continuación se describe brevemente este fenómeno físico en el cual se basa el método de Aprendizaje de la Ordenación propuesto en el Capítulo 2.

1.4.1 Descripción general

Las supernovas, fenómenos estelares de gran luminosidad, han causado curiosidad desde la antigüedad. El estudio del fenómeno y las diversas técnicas desarrolladas por la astrofísica han permitido determinar que las supernovas son el destino final de algunas estrellas. Inicialmente, las estrellas queman su combustible, y este, a su vez, se convierte en elementos de mayor masa: del hidrógeno al helio, luego carbono, y así sucesivamente hasta llegar a un núcleo de hierro, ordenados por capas como una cebolla. Las altas temperaturas y presiones internas descomponen los átomos de hierro haciendo que el núcleo sea inestable y colapse en cualquier momento, dando lugar a una supernova. Las ondas de choque generadas internamente por este colapso provocan una expulsión de fotones, neutrones y material articulado. [38]

Las partículas expulsadas interactúan entre ellas y con otros cuerpos cercanos a su trayectoria, donde las partículas y cuerpos de mayor masa atraen a las que tienen menor masa a medida que pierden su impulso. [39]

Haciendo una breve síntesis del fenómeno, supernova sería una explosión de partículas que se organizan en el tiempo de acuerdo a su velocidad y masa siguiendo las leyes de la conservación de la energía. Partiendo de esto se podrían delinear algunas reglas que logren encontrar buenas respuestas a problemas de optimización.

La explosión inicial se puede emular como aleatoriedad y la interacción de partículas expulsadas restringirse a dos interacciones: una que expande y otra que atrae las partículas. Aunque es un fenómeno sumamente complejo, por simplicidad, se tomarán sólo dos de las fuerzas presentes en el mismo: el impulso y la gravitación; la interacción entre ambas fuerzas llena las expectativas de expansión y atracción mencionadas. [40]

1.5 Fundamentación de la metodología utilizada.

Para el desarrollo de la investigación se emplea una metodología mixta ya que se utilizan elementos de la metodología cuantitativa y cualitativa. El enfoque cualitativo permite la interpretación del fenómeno y el enfoque cuantitativo, su

explicación. Esta es una tendencia que se está utilizando cada vez con mayor frecuencia en las investigaciones.

La investigación cualitativa propone la estancia prolongada en el campo y la observación persistente de los focos principales de la investigación. Se recogen las informaciones y luego se analizan desde distintos ángulos a fin de contrastarlos, realizando el cruzamiento e interpretando y hallando las coincidencias de los resultados alcanzados.

Se consideran de utilidad los métodos cuantitativos, ya que los números constituyen instrumentos de modelación de la realidad, de aproximación al conocimiento y contribuyen a ordenar la información en el proceso de producción del conocimiento.

La metodología mixta posibilita aprovechar las tendencias, la direccionalidad que pueden mostrar algunos métodos cuantitativos sobre la base del análisis cualitativo, para cruzar y verificar la información obtenida aplicando diferentes métodos. Por eso se considera que la aproximación a la verdad está vinculada con la pluralidad metodológica, ya que el método debe poseer la capacidad de reflejar el movimiento, la dinámica, la esencia del objeto y esto no se logra sin la participación compensatoria de los diversos métodos existente. [41]

Entre los métodos cualitativos y cuantitativos empleados en este trabajo se encuentran la modelación algorítmica, la experimentación y el análisis estadístico.

Los métodos científicos, según su etimología, son el camino hacia el conocimiento. En la presente investigación se utilizan varios de estos métodos:

- Nivel teórico.
  - ✓ Inducción – deducción: La inducción expresa el movimiento de lo particular a lo general, o sea se llega a generalizaciones partiendo del análisis de casos particulares, mientras la deducción expresa el movimiento de lo general a lo particular. La combinación de ambos permite estructurar el conocimiento científico obtenido en la revisión bibliográfica.
  - ✓ Histórico – lógico: El método lógico debe basarse en los datos que proporciona el método histórico, de manera que no se convierta en un

simple razonamiento especulativo. De igual forma, lo histórico no debe limitarse a la simple descripción de los hechos, sino explicarlos a partir de la lógica de su desarrollo [42]. Este método da la posibilidad de conocer el problema de la ordenación en la Recuperación de Información en su origen y desarrollo, y proyectar el análisis de la evolución histórica de los métodos de Aprendizaje de la Ordenación con la lógica de su comportamiento futuro.

- ✓ Análisis y síntesis: A través del análisis se distinguen y revisan por separado los elementos relacionados con el Aprendizaje de la Ordenación para la R.I. Partiendo de esto, se emplea la síntesis para establecer nexos, comparar resultados, determinar características comunes y aspectos distintivos de los diferentes enfoques estudiados, lo que permite arribar a conclusiones.
- ✓ Comparación: Para establecer las similitudes y diferencias entre los modelos y categorías existentes dentro del Aprendizaje de la Ordenación para la R.I.

- Nivel matemático y estadístico.
  - ✓ Métodos de la Estadística: Con la utilización de la estadística descriptiva (distribución de frecuencias, medidas de tendencia central y medidas de la variabilidad) y las pruebas no paramétricas para muestras relacionadas o pareadas, es posible evaluar el desempeño del método RankSupernova y comparar sus resultados con los métodos referenciados en la literatura.

1.6 Tendencias y herramientas de desarrollo.

Para la implementación del método de Aprendizaje de la Ordenación RankSupernova se utilizó Java como lenguaje de programación, debido a que es multiplataforma y cumple con todos los requerimientos necesarios para desarrollar nuevos modelos. Además, se piensa incorporarlo a L2RLab [43], un entorno integrado de experimentación para la tarea del Aprendizaje de la Ordenación. Otro motivo es lograr uniformidad con la comunidad científica ya que la mayoría de los

métodos desarrollados están programados en Java o se pretenden programar en este lenguaje.

Java se está moviendo rápido al plano de los lenguajes tradicionales de inteligencia artificial como Lisp y Prolog. Hay ya varios programas y APIs en Java para técnicas de *SoftComputing* y aprendizaje automático, de libre distribución y disponible en línea como GPL (*open source*). Esto muestra la popularidad explosiva de Java en el campo de inteligencia artificial y *SoftComputing*. [44]

Como IDE (entorno integrado de desarrollo) de programación se emplea Eclipse, que definido por la Fundación del mismo nombre es: "*una especie de herramienta universal - un IDE abierto y extensible para todo y nada en particular*". La versión que se utiliza en este trabajo es portable, por lo que a la ventaja de ser libre, se le suma la posibilidad de estar ejecutando a la vez varias versiones del método propuesto lo que permite ahorrar tiempo en la etapa de experimentación, ajuste de parámetros y pruebas con las distintas colecciones de datos.

1.7 Discusión y trabajos futuros.

Como trabajos futuros vinculados a la presente investigación están:

- ✓ Conducir más experimentos con nuevas y referenciadas colecciones de datos de mediano y gran tamaño para validar la eficiencia, y evaluar el desempeño del método RankSupernova bajo disímiles condiciones.
- ✓ Utilizar otras medidas de desempeño, por ejemplo, P@n y NDCG; como función general *E*, con el objetivo de evaluar el comportamiento de la precisión en la ordenación, considerando el mismo diseño experimental conformado para MAP.
- ✓ Aplicar otras variantes de error en la función de pérdida básica, como por ejemplo, el Error Cuadrático Medio.
- ✓ Rediseñar la función de *ranking* considerando las relaciones y el comportamiento de los rasgos.
- ✓ Emplear algoritmos adaptativos para tratar el problema de ajustar los parámetros de RankSupernova. Debido a que ajustar los parámetros de

forma empírica (manualmente) es usualmente difícil, especialmente cuando hay muchos parámetros y las medidas de evaluación no son uniformes.

- ✓ Desarrollar estrategias multi-supernovas basadas en la concepción de RankSupernova, con la intención de mejorar el rendimiento y disminuir el tiempo computacional.

En sentido general, sería interesante poder diseñar modelos de Aprendizaje de la Ordenación que implementasen algoritmos de aprendizaje *online*, que se adapten a los cambios que ocurren en la Web a tiempo real.

Por otro lado, la tendencia en LETOR resulta en proponer nuevos conjuntos de entrenamiento de gran tamaño (p.e: MSLR-WEB10K[1]), cuya manipulación reviste un reto para muchos modelos de aprendizaje. En tal caso, sería oportuno utilizar elementos de la computación paralela, el aprendizaje de conjunto y llevar a cabo aproximaciones a algoritmos existentes que reduzcan el costo y mantengan la precisión.

Incorporar nuevas estrategias a los métodos de Aprendizaje de la Ordenación para evadir el sobre-ajuste y el estancamiento de las soluciones en mínimos locales, también constituye una tarea en la que se debe enfocar la comunidad de investigadores.

Finalmente como pasó en su momento que la inserción de juicios de relevancia dados por especialistas, reformuló el camino investigativo de los modelos de Aprendizaje de la Ordenación; ahora, avizorando un futuro emergente, sería oportuno pensar en la concepción de nuevos modelos que tomasen en consideración no sólo las consultas, los documentos recuperados para tales consultas y los juicios de relevancia, sino además, el contexto en el que se formulan las consultas.

1.8 Conclusiones

En el presente capítulo se puede constatar como los métodos tradicionales de Recuperación de Información, han ido cediendo su

[1] Disponible en: http://research.microsoft.com/en-us/projects/mslr/

espacio a nuevas técnicas de aprendizaje automático, debido a la necesidad imperante de afrontar nuevos retos y problemas de ordenación que exigen mayor precisión para lograr la satisfacción del usuario.

Es importante resaltar, después de haber analizado cada categoría, que en los últimos años se observa una tendencia hacia los métodos que optimizan directamente medidas de desempeño, pues estos brindan elementos teóricos y resultados prometedores en cuanto al rendimiento alcanzado en la ordenación con respecto a las propuestas tradicionales.

A pesar de estos avances, los métodos desarrollados muestran dificultad para alcanzar una aproximación global y precisiones estables a nivel de consulta. Todo lo cual dirige su atención a utilizar nuevas técnicas propias o híbridas de *SoftComputing*; en tal caso, Supernova es un algoritmo de optimización global que puede dar respuesta a esta problemática.

Para el desarrollo de esta investigación se emplea una metodología mixta, aprovechando las ventajas que brinda la integración de los métodos cualitativos y cuantitativos.

Capítulo 2: Supernova en el Aprendizaje de la Ordenación

2.1 Introducción

En este capítulo se describen las principales características del fenómeno físico de la Supernova y su modelación algorítmica. Este algoritmo se adapta al problema del Aprendizaje de la Ordenación, proponiéndose un nuevo método que permite optimizar directamente las medidas de desempeño utilizadas en el Aprendizaje de la Ordenación para la R.I.

2.2 Optimización global con Supernova

Como se explicó en el epígrafe 1.4 las partículas producidas por la explosión de la supernova interactúan entre ellas y llegan a un estado de equilibrio y orden donde las partículas de menor masa son atraídas por otras de mayor masa de acuerdo a la distancia entre ellas.

Una heurística que emule las etapas del fenómeno de las supernovas podría encontrar regiones importantes dentro de una región de búsqueda pues las soluciones se aglomerarían alrededor de mejores soluciones por la atracción que ejercen estas sobre otras no tan buenas, tal como sucede con las partículas expulsadas de las supernovas. [37]

2.2.1 Descripción del fenómeno físico

Supernova es una explosión de partículas que se organizan en el tiempo de acuerdo a su velocidad y masa siguiendo las leyes de la conservación de la energía. Varias fuerzas están presentes en este fenómeno y definen la interacción de atracción y expansión entre las partículas. Para elaborar el algoritmo de optimización se toman dos de estas fuerzas: el impulso y la energía potencial gravitatoria.

El impulso p hará que la partícula intente seguir con la dirección y velocidad que traía, de acuerdo a:

$$p = m\vec{v} \tag{2.1}$$

donde m representa la masa y $\vec{v}$ la velocidad de la partícula.

La energía potencial gravitatoria F de un grupo de *k* partículas, clásicamente está descrita por:

$$F = m\sum_{i=1}^{k}\frac{GM_i}{r_i^2}\vec{u} \tag{2.2}$$

donde m representa la masa de la partícula para la cual se calcula la energía potencial gravitatoria; M_i es la masa de la otra partícula, G es la constante de gravitación universal, r_i representa la distancia entre ambas y $\vec{u}$ es un vector unitario que da la dirección entre ambas partículas.

La interacción gravitacional intentará atraer las partículas de menor masa hacia las de mayor masa de acuerdo a sus distancias variando la dirección y velocidad inicial dada por la explosión. La masa es directamente proporcional a la fuerza, por lo tanto, entre mayor masa tenga una partícula mayor será su atracción hacia otras, mientras que la distancia es inversamente proporcional, es decir, las partículas más lejanas tendrán menos inferencia en la dirección del movimiento que las más cercanas. [39]

En términos de optimización, si se define la masa como el valor de la función objetivo evaluada en la solución actual, donde una mayor masa es una mejor respuesta respecto a otra de menor masa, sería deseable tener una interacción donde las partículas con mayor masa atrajeran a otras partículas de menor masa como en el fenómeno de la supernova. Las ecuaciones planteadas, a pesar de tratarse de una simplificación del fenómeno original, llegan a un equilibro entre la fuerza atrayente (gravedad) y la fuerza detractora (impulso) mostrando el mismo comportamiento del fenómeno.

En la Figura 2.1, se muestra la estrategia de búsqueda deseada para la heurística, donde las estrellas representan las soluciones encontradas para la función delimitada que tiene su mínimo en (0,0). En la parte (a) se observan los puntos aleatorios uniformemente distribuidos en el espacio de búsqueda. En la parte (b) se muestra una pequeña expansión inicial y luego en la parte (c) se muestra la trayectoria de atracción esperada. Luego de llegar al equilibrio se toma el mejor punto, identificado por el cuadrado en la parte (d) de la Figura, y a partir de este

punto, se explotan nuevos puntos con un radio de búsqueda más reducido (elipse punteada).

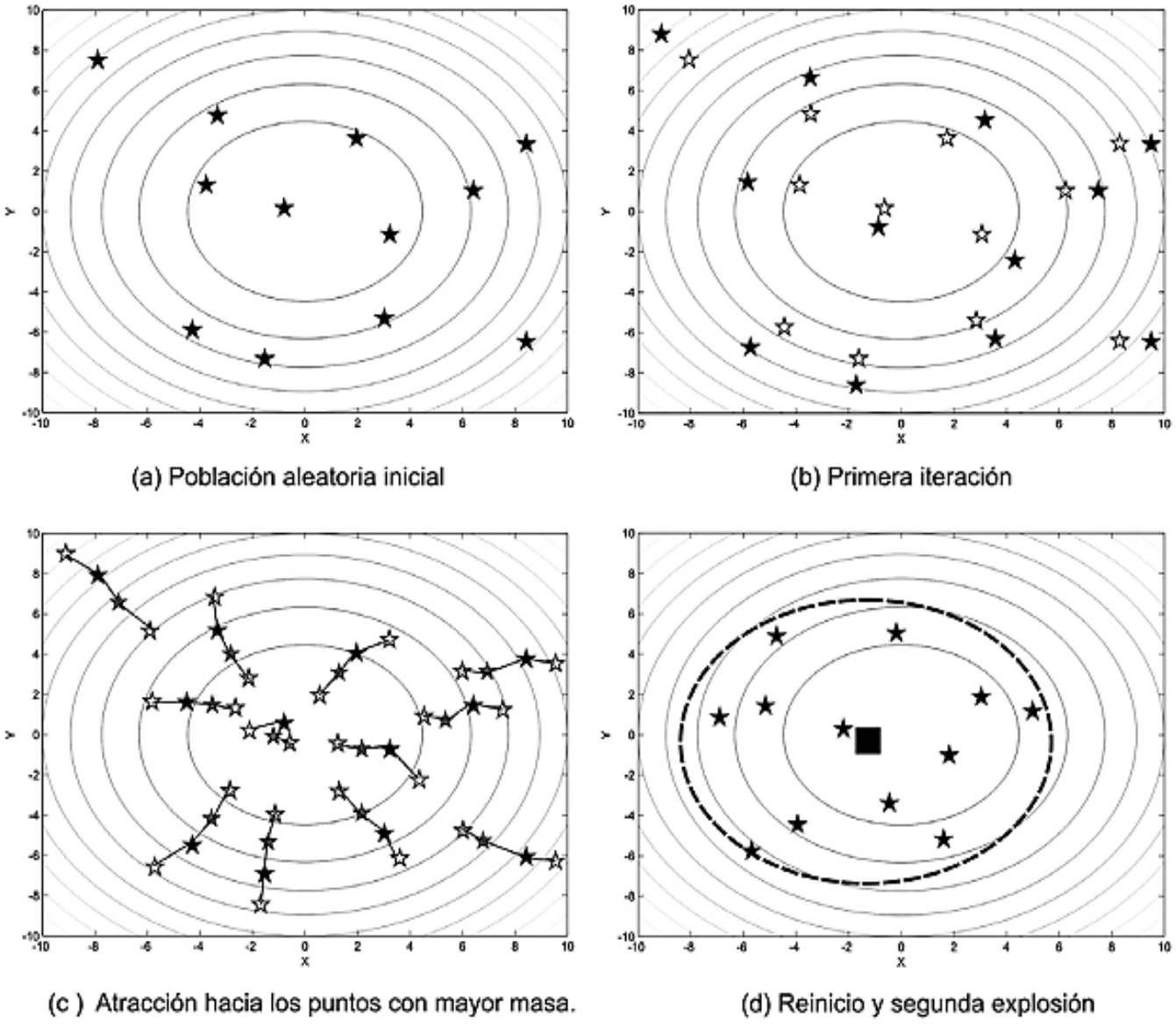


Figura 2.1: Distribución de partículas para la función esférica [37]

En la Tabla 2.1 se establece el paralelo entre algunas partes del fenómeno físico y su aplicación en el modelo de optimización. Se debe tener en cuenta que para una minimización la masa sería el inverso de la función objetivo, por lo tanto, es importante evitar que la función objetivo sea cero, por esto se emplea su valor

escalado. La velocidad se toma aleatoria, y el potencial gravitacional y la energía cinética son las formas de interacción entre partículas.

Tabla 2.1 Equivalencias entre el modelo físico y el modelo de optimización

Modelo Físico	Modelo de optimización
Partícula	Solución $x \in N$ (N: conjunto de soluciones posibles)
Masa	Valor de x en la función objetivo (escalado): $\varphi(S(x))$
Velocidad	Paso entre solución y solución
Potencial gravitacional	Movimiento de la partícula hacia la mejor solución
Impulso	Movimiento de la partícula en su trayectoria original

2.2.2 Estrategia general

En el proceso de elaboración del algoritmo de optimización, se divide el fenómeno físico explicado anteriormente en cuatro fases:

- ✓ Explosión

  Para la explosión de partículas se generan valores aleatorios. Existen dos formas de realizar esta explosión. Se denomina de Tipo I cuando, a partir de un punto x_0, se generan *k* soluciones aleatorias (parte *d* de la Figura 2.1). Como no siempre se dispone de una solución inicial, sino de un espacio de búsqueda, en la explosión Tipo II, se generan *k* soluciones aleatorias distribuidas sobre una región *A* (parte *a* de la Figura 2.1).

- ✓ Escalamiento

  Cada una de las soluciones generadas en la explosión se evalúa respecto a la función objetivo *f* obteniéndose un número perteneciente a los reales. Al analizar la ecuación (2.2) se puede observar que las variables son: masa y distancia; ambas magnitudes positivas. El uso de números negativos para la masa podría generar falsos movimientos en la interacción y no llegar a una buena solución. En el caso de la minimización, al usar el inverso de la función como masa, estaría indefinido para el cero. Por tanto, para el correcto funcionamiento de la heurística es necesario garantizar que el valor

de la función objetivo no sea negativo ni cero, para esto se utiliza un escalamiento dinámico lineal modificado.

- ✓ Cálculo de fuerzas atractivas y repulsivas

  Una vez que se tienen las soluciones con sus respectivos valores de masa se inicia la interacción entre ellas. Se halla la dirección de los vectores unitarios de trayectoria para cada partícula, en el caso de inicio Tipo I, desde el punto de inicio x_0 hacia un punto cualquiera; y para el caso de inicio Tipo II, desde el mejor punto encontrado hacia los otros puntos aleatorios [37]. Con estos vectores se hallan las distancias r_i entre los puntos. Para cada partícula se calcula la dirección y la distancia a la que se moverá, de acuerdo al impulso y la atracción que ejercen las otras partículas.

  La Figura 2.2 representa la interacción entre partículas; r_{CA}, r_{BC} y r_{AB} son las distancias, los vectores $\vec{d}$ son los vectores del impulso, y los vectores $\vec{u}$ son los vectores correspondientes a la atracción gravitatoria.

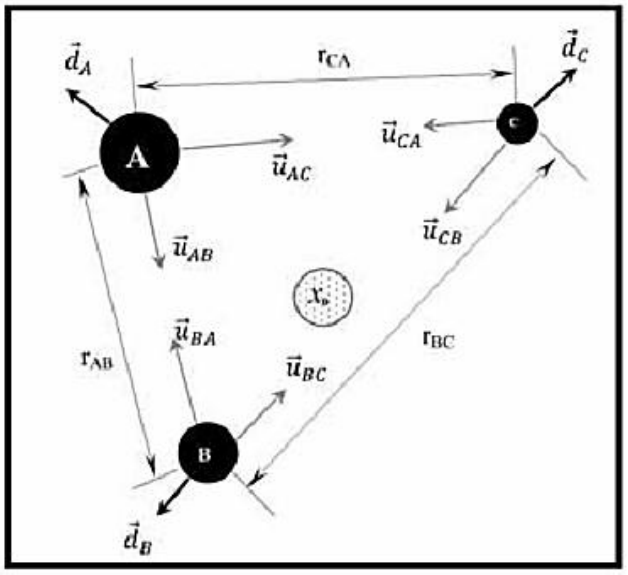


Figura 2.2: Vectores de la interacción entre las partículas [37]

Los nuevos puntos se obtienen con la sumatoria de los vectores de impulso y atracción gravitatoria hallados, y comienza la siguiente iteración que

consiste en volver a calcular las fuerzas del impulso y la gravedad. Este proceso se repetirá hasta llegar a un número de iteraciones determinado.

- ✓ Reinicialización

  Para mejorar la calidad de la respuesta, es necesario refinar la búsqueda. Es por esto que se toma la mejor partícula encontrada y se realiza una explosión Tipo I, reduciendo el radio de alcance de la explosión y con esto la región de búsqueda.

2.2.3 Algoritmo clásico

Siguiendo la estrategia explicada en el epígrafe anterior, se elabora el algoritmo de optimización y se plantean las ecuaciones que permiten simular el comportamiento de las partículas en el fenómeno físico.

- ✓ Explosión

Para la explosión Tipo I se genera el conjunto $S(0)$ como se muestra en la ecuación (2.3). Sobre una solución inicial x_0 se generan *k* puntos aleatorios, una fuerza de explosión λ y una velocidad v para cada partícula, que son parámetros independientes; $\vec{w}$ es un vector unitario aleatorio que prescribe la dirección de cada partícula.

$$S(0) = \{\overrightarrow{x_0} + \lambda v\overrightarrow{w_1}, \ldots, \overrightarrow{x_k} + \lambda v\overrightarrow{w_k}\} \tag{2.3}$$

Para la explosión Tipo II el conjunto inicial $S(0)$ se genera al distribuir uniformemente los *k* puntos sobre una región acotada *A*, donde cada s es una solución aleatoria del espacio de búsqueda *A*.

$$S(0) = \{\overrightarrow{s_1}, \ldots, \overrightarrow{s_k}\} \tag{2.4}$$

- ✓ Evaluación

Una vez generado el conjunto $S(0)$ se evalúa la función objetivo escalada $\varphi(S(0))$ para cada partícula. En este caso se evalúa el conjunto $S(0)$ con la función objetivo $f(S(0))$, y se encuentra el valor mínimo m_{min} del conjunto y se escala de acuerdo a:

$$\varphi(S(0)) = f(S(0)) - (m_{min} - 1) \tag{2.5}$$

✓ Cálculos

Las partículas del conjunto $S(t)$, siendo t una iteración cualquiera, se vectorizan. La distancia se normaliza por medio de un promedio ponderado, donde los puntos más cercanos tendrán un valor mayor, mientras los valores más lejanos tendrán valores cercanos a cero, conservando el comportamiento de la distancia en (2.2), donde las partículas ejercen más atracción a los puntos cercanos, mientras tienen una menor incidencia en el comportamiento de las partículas más lejanas. El movimiento de cada partícula se halla según:

$$S'(t) = \frac{F}{\varphi(S(t))}\,\vec{d} + \frac{G}{\varphi(S(t))}\sum_{i=1}^{k}\frac{1-\frac{r_i}{\sum_{j=1}^{k} r_j}}{\varphi(S_i(t))}\,\vec{u} \qquad (2.6)$$

donde la primera parte corresponde a la energía cinética y la otra al potencial gravitatorio. F y G son parámetros constantes, *r* es la distancia entre un punto *i* y el que se está evaluando. $\vec{d}$ y $\vec{u}$ son vectores unitarios de dirección que en el primer caso señala la dirección que traía la partícula y en el segundo caso la dirección de cualquier solución hacia otra solución *i* cualquiera.

En la Figura 2.2 se mostró para un sistema de tres partículas las diferentes relaciones entre ellas, desde una partícula inicial x_0, se hallan 3 partículas aleatorias $S(0) = \{A, B, C\}$; para la partícula A se aplica la ecuación (2.6) quedando como sigue:

$$S'_A(1) = \frac{F}{\varphi_A}\,\overrightarrow{d_A} + \frac{G}{\varphi_A}\left(\frac{1-\frac{r_{AB}}{r_{AB}+r_{AC}}}{\varphi_B}\,\vec{u}_{AB} + \frac{1-\frac{r_{AC}}{r_{AB}+r_{AC}}}{\varphi_C}\,\vec{u}_{AC}\right) \qquad (2.7)$$

✓ Reinicio

Luego de realizar la cantidad de movimientos establecidos, se escoge la mejor partícula encontrada (cuyo valor de masa es el mínimo) y a partir de esta se realiza el reinicio mediante una explosión Tipo I disminuyendo el valor de λ con un factor ∝, como se muestra en la siguiente ecuación.

$$\lambda_{R+1} = \frac{\lambda_R}{\propto} \qquad (2.8)$$

Con esta disminución se logra que la explosión tenga menos fuerza en cada reinicio, explotando así, la mejor zona encontrada en el reinicio anterior. Este proceso se repite hasta llegar al criterio de parada o al número de reinicios planteado.

En el Pseudocódigo 2.1 se muestra la secuencia general seguida por el algoritmo Supernova en el proceso de optimización; se inicializa con una explosión de k elementos como se describió anteriormente. En este caso, se pueden utilizar ambos métodos de inicio indistintamente: tipo I o tipo II. Luego se evalúa la función de masa y se inicializa la matriz de mejores elementos *M*. Posteriormente, se inicia la etapa de cálculo, se evalúa nuevamente y se actualizan las matrices *M* y *Old*, la matriz *Old* es la matriz donde se guardan las soluciones anteriores y permite la vectorización de los puntos. Cuando se llega al número de iteraciones *T* definido, se vuelve a explotar sobre el mejor punto encontrado y se repite el proceso hasta llegar al número de reinicios estipulados: *RR*. La matriz *M* conserva los mejores puntos encontrados por cada partícula. [37]

El algoritmo requiere como datos de entrada: la función a optimizar, un punto inicial o los bordes específicos de la región de búsqueda (de acuerdo al método de inicialización que se emplee), dimensiones y los parámetros siguientes:

- *k*, *T*, *RR*: son el número total de partículas que interactúan, el número de iteraciones máximas y el número de reinicios máximos respectivamente.
- *F* y *G*: son constantes que representan los pesos que tienen el impulso y la fuerza gravitatoria respectivamente.
- λ y $\propto$: La fuerza de explosión y su tasa de disminución en el tiempo, tienen que ver con el tamaño del área a explorar y la precisión que se requiera.

Pseudocódigo 2.1: Algoritmo Supernova

t=0; S=0; M=0; R=0
mientras $R \leq RR$ || criterio de parada **hacer**
    Explotar: $S(t)$ //ecuación 2.3 o 2.4
    Evaluar: $\varphi(S(t))$
    $M = S(t)$
    mientras $t \leq T$ || criterio de parada **hacer**
        Calcular: $S(t+1) = (S(t) + S'(t))$ //ecuación 2.6
        Evaluar: $\varphi(S(t+1))$
        M_k = min (M_k | $S(t+1)$) //guarda el mínimo de cada partícula
        $Old = S(t-1)$
        $t = t+1$
    fin mientras
    $\lambda_{R+1} = \frac{\lambda_R}{\propto}$ //ecuación 2.8
    $R = R+1$
fin mientras

2.3 Método Propuesto. RankSupernova

2.3.1 Formulación general

Supongamos que $Q = \{q_1, q_2, \dots, q_m\}$ es el conjunto de consultas en el entrenamiento, D constituye la colección documental y que $Y = \{r_1, r_2, \dots, r_k\}$ es el conjunto de juicios de relevancia. Un juicio de relevancia constituye una medida de cuan bueno es un documento recuperado con respecto a las necesidades de un usuario, reflejadas en la consulta formulada.

Cada consulta q_i (representada como una lista de términos $\{t_1, t_2, \dots, t_{h(q_i)}\}$ donde $h(q_i)$ denota la cantidad de términos para la i-ésima consulta) está asociada con una lista de documentos recuperados $d_i = \{d_{i_1}, d_{i_2}, \dots, d_{i_{n(q_i)}}\}$ y una lista de etiquetas $y_i = \{y_{i_1}, y_{i_2}, \dots, y_{i_{n(q_i)}}\}$, donde $n(q_i)$ denota el tamaño de la lista; $d_i \subseteq D$ e $y_i \subseteq Y$ para la consulta $q_i \in Q$, $d_{ij} \in d_i$ indica el j-ésimo documento en d_i, así como $y_{ij} \in y_i$ denota la etiqueta del documento d_{ij}. De esta manera cada documento d_{ij} está representado como una lista de rasgos o características.

Finalmente el conjunto de entrenamiento queda definido como sigue:

$$S = \{(q_i, d_i, y_i)\}_{i=1}^{m}$$

El ordenamiento se concentra en obtener una predicción para una consulta dada q_i y la lista de documentos asociados d_i, usando el modelo de ordenación.

El modelo de ordenación, denotado como f, es una función a nivel de documento, la cual constituye una combinación lineal de los rasgos en un vector de rasgos $\emptyset(q_i, d_{ij})$:

$$f(q_i, d_{ij}) = w^T \emptyset(q_i, d_{ij}) \quad (2.9)$$

donde w denota el vector de peso y $\emptyset(q_i, d_{ij})$ es el vector de rasgos creado para cada par consulta-documento (q_i, d_{ij}), donde $i = 1, 2, \dots, m$ y j $= 1, 2, \dots, n(q_i)$. En la ordenación para la consulta q_i se asigna una puntuación para cada uno de los documentos usando la función $f(q_i, d_{ij})$ y ordenando posteriormente tales documentos en correspondencia con las puntuaciones que le fueron dadas.

Así se obtiene una predicción denotada como π_i, que es la predicción realizada por el modelo de ordenación sobre d_i en términos de la consulta q_i. Se emplea Π_i para denotar el conjunto de todas las predicciones posibles sobre d_i, y se usa $\pi_i(j)$ para denotar la posición del elemento j, por ejemplo d_{ij}. El ordenamiento se concentraría en obtener una predicción $\pi_i \in \Pi_i$ para una consulta dada q_i y la lista de documentos asociados a d_i, usando el modelo de ordenación.

En la R.I., las medidas de desempeño son usadas para evaluar la efectividad de un modelo de ordenación, el cual usualmente está basado en consultas. Esto significa que la medida se define sobre una lista de documentos ordenados con respecto a una consulta. Entre las medidas de desempeño más empleadas están: MAP (*Mean Average Precision*) [17], NDCG (*Normalized Discounted Cumulative Gain*) [18], y P@n (*Precision @t n*) [17]. Para un análisis más detallado, ver el apartado 3.3 Medidas de Evaluación.

En este trabajo se utiliza una función general $E(\pi_i, y_i) \in [0, +1]$ para representar las medidas de evaluación. El primer argumento de E es la predicción π_i creada usando el modelo de ordenación. El segundo argumento y_i es la lista de juicios que determinan el orden ideal. E mide la correspondencia entre π_i e y_i. La mayoría de las medidas de desempeño retornan valores reales entre $[0, +1]$.

La predicción ideal puede ser denotada como π_i^*.

Ahora, como puede existir más de una predicción ideal para una consulta, se denota Π_i^* como el conjunto de posibles predicciones ideales para la consulta q_i. Por tanto $\pi_i^* \in \Pi_i^*$ y se tiene que $E(\pi_i^*, y_i) = 1$.

En la Tabla 2.2 se presenta un resumen de las notaciones descritas anteriormente.

Tabla 2.2 Resumen de notaciones para el modelo de Aprendizaje de la Ordenación

Notación	Descripción
$q_i \in Q$	Consulta
$d_i = \left\{d_{i_1}, d_{i_2}, \dots, d_{i_{n(q_i)}}\right\}$	Lista de documentos recuperados para q_i
$d_{ij} \in d_i$	j-ésimo documento en d_i
$y_i = \left\{y_{i_1}, y_{i_2}, \dots, y_{i_{n(q_i)}}\right\}$	Lista de juicios para d_i con respecto a q_i
$y_{ij} \in \{r_1, r_2, \dots, r_k\}$	Juicio de d_{ij} con respecto a q_i
$S = \{(q_i, d_i, y_i)\}_{i=1}^{m}$	Conjunto de entrenamiento
$\pi_i \in \Pi_i$	Predicción del modelo de ordenación para q_i
$\pi_i^* \in \Pi_i^*$	Predicción ideal para q_i
$\emptyset(q_i, d_{ij})$	Vector de rasgos para (q_i, d_{ij})
f	Modelo de ordenación
$E(\pi_i, y_i) \in [0, +1]$	Evaluación de π_i con respecto a y_i para q_i

Idealmente, el objetivo del modelo de ordenación es maximizar la precisión alcanzada sobre un conjunto de entrenamiento, en términos de una medida de desempeño de R.I.

Para ello, podemos plantear la minimización de una función de pérdida básica definida en [27] como sigue:

$$\boldsymbol{R(f) = \sum_{i=1}^{m}(E(\pi_i^*, y_i) - E(\pi_i, y_i)) = \sum_{i=1}^{m}(1 - E(\pi_i, y_i))} \quad (2.10)$$

donde π_i es la predicción determinada por el modelo de ordenación para la consulta q_i.

2.3.2 Algoritmo

Teniendo en cuenta las notaciones antes mencionadas, se adapta el algoritmo Supernova al Aprendizaje de la Ordenación y se propone un nuevo método: RankSupernova (Pseudocódigo 2.2).

Pseudocódigo 2.2: Método RankSupernova

Entrada: $S = \{(q_i, d_i, y_i)\}_{i=1}^{m}$ y los parámetros: *T, RR y E*

Desde *R=0* hasta *RR* **hacer**
  Explotar: $partículas(t)$ //ecuación 2.4 (cuando R=0) o ecuación 2.3
  Evaluar: $masa(t)$
  Actualizar mejor: *best* //guarda la mejor partícula
  Desde *t=0* hasta *T* **hacer**
    Calcular movimiento:
      $particula(t+1) = partícula(t) + S'(t)$ //ecuación 2.6
    Evaluar: $masa(t+1)$
    Actualizar mejor: *best*
    $t = t + 1$
  fin desde
  $\lambda_{R+1} = \frac{\lambda_R}{\propto}$ //ecuación 2.8
  $R = R + 1$
fin desde
Construir el modelo de ordenación *f* con la partícula *best*
Salida: *f*

A los parámetros que forman parte de los datos de entrada y que fueron definidos en la sección 2.2.3, se añade la función a optimizar (*E*), que de forma general corresponde con una de las medidas de desempeño de la R.I.

El método inicia con una explosión de Tipo II (ecuación 2.4) y en los restantes reinicios se utiliza una explosión Tipo I (ecuación 2.3) a partir de la mejor partícula encontrada (*best*).

Como en el caso del Aprendizaje de la Ordenación solo es de interés obtener la mejor partícula, se sustituye la matriz *M* empleada en el Pseudocódigo 2.1 por la mejor partícula (*best*).

En el cálculo del movimiento, específicamente para obtener el valor del potencial gravitacional a partir de la interacción entre las partículas, se cambia el patrón original y se utilizan los valores actualizados de las partículas que se hayan modificado, así se agiliza la obtención de mejores soluciones.

Finalmente el algoritmo devuelve su mejor partícula, con la cual se construye la función lineal ponderada, que corresponde a la función de ordenación aprendida sobre la base del conjunto de entrenamiento y la medida de desempeño utilizada.

2.4 Conclusiones

En este capítulo se presentó RankSupernova, un novedoso método de Aprendizaje de la Ordenación, que es capaz de optimizar directamente cualquier medida de desempeño de la R.I.

Este nuevo método se basa en el algoritmo Supernova que conserva los patrones generales del fenómeno físico de las supernovas: la explosión, el movimiento repulsivo y la atracción hacia los elementos de mayor masa.

La analogía con el movimiento de las partículas se mantiene como base principal, pero el algoritmo propuesto utiliza los valores actualizados de las partículas que se hayan modificado, con la intensión de acelerar la convergencia del método a mejores soluciones.

Capítulo 3: Resultados experimentales

3.1 Introducción

En este capítulo se presentan los resultados experimentales de la evaluación del método propuesto en términos del rendimiento (precisión) alcanzado en la ordenación para los conjuntos de datos de LETOR[2] 3.0, tomados de las colecciones de OHSUMED y Gov. Se consideraron como criterios para evaluar el desempeño del algoritmo en la ordenación, las medidas P@n, MAP y NDCG, ampliamente utilizadas en la R.I.

Este estudio experimental se basó en una comparación directa con los principales métodos que establecen actualmente el estado del arte del Aprendizaje de la Ordenación. Finalmente, se tomaron los resultados arrojados por cada algoritmo considerado y se realizaron pruebas paramétricas utilizando un modelo estadístico de comparación de medias para muestras relacionadas.

3.2 Colecciones de datos utilizadas

Con el creciente desarrollo de modelos para el Aprendizaje de la Ordenación, se hizo necesario crear conjuntos de datos de referencia que pudieran usarse en la comparación de algoritmos de aprendizaje existentes y en la evaluación de los nuevos modelos que surgían.

Para tratar este problema Microsoft Research Asia[3] propone LETOR. Los conjuntos de datos de LETOR se derivan de las colecciones de datos existentes y ampliamente usadas en R.I., estas son: OHSUMED y Gov. Letor es un estándar de referencia para la investigación en Aprendizaje de la Ordenación [45]. Contiene los rasgos típicos, los juicios de relevancia, las particiones de datos, las herramientas de evaluación y varios métodos de Aprendizaje de la Ordenación de referencia.

[2] Disponible en: http://research.microsoft.com/en-us/um/beijing/projects/letor/
[3] Disponible en: http://research.microsoft.com/en-us/labs/asia/default.aspx

La versión 1.0 fue lanzada en Abril de 2007, la 2.0 en Diciembre de ese mismo año y en Diciembre de 2008, la versión 3.0. En esta investigación se utilizan las colecciones OHSUMED y Gov de la versión 3.0 de LETOR.

Estas colecciones de datos contienen consultas, el contenido de los documentos recuperados, y juicios humanos de relevancia de los documentos con respecto a las consultas. Con vistas a la tarea del Aprendizaje de la Ordenación se extrajeron rasgos de estos conjuntos de datos, incluyendo rasgos convencionales (frecuencia de término, frecuencia inversa del documento, longitud del documento, BM25, y modelo de lenguaje para R.I.) y nuevos rasgos propuestos por SIGIR[4] (*HostRank*, propagación del rasgo y *PageRank*). Finalmente se empaquetaron en LETOR estos rasgos extraídos, las consultas y los juicios de relevancia. [17]

Con la intensión de potenciar una comparación directa entre modelos, se prefijó un esquema de experimentación, donde fue considerada una validación cruzada con 5 particiones, incluyendo en cada partición un subconjunto de entrenamiento, validación y prueba.

3.2.1 LETOR OHSUMED

La colección OHSUMED fue creada para la investigación en R.I. Esta es un subconjunto de Medline, una base de datos sobre publicaciones médicas. La colección consta de 348 566 registros de 270 revistas médicas, correspondientes al período de 1987-1991. [17]

En OHSUMED hay 106 consultas sobre necesidades de búsqueda médica y cada una de estas tiene un número de documentos asociados. El grado de relevancia de los documentos con respecto a las consultas es dado por humanos que los agrupan en tres niveles: definitivamente relevante “2”, parcialmente relevante “1” o no relevante “0”. Existen un total de 16 140 pares de consulta-documento etiquetados con juicios de relevancia, teniendo en cuenta 45 rasgos.

[4] Special Interest Group on Information Retrieval, www.sigir.org

3.2.2 LETOR Gov

En TREC[5] 2003 y 2004, hay una línea peculiar de investigación para la recuperación de información en la Web, denominada la huella de la Web. La meta de esta línea es estudiar el comportamiento de la recuperación cuando la colección a ser investigada está en una estructura de grandes hipervínculos como Internet. Las huellas de la Web usan la colección Gov, que es basada en una selección del dominio ". gov" realizada en enero de 2002. Hay en total 1 053 110 documentos html en esta colección, junto con 11 164 829 hipervínculos. [17]

La destilación del tema es una tarea que permite encontrar puntos de entrada a aquellos sitios Web que están principalmente dedicados al tema de la consulta. El objetivo es devolver la página principal del sitio en lugar de las páginas que contengan alguna información, teniendo en cuenta que las páginas iniciales proporcionan una mejor apreciación global del contenido del sitio. Los asesores humanos utilizan juicios binarios para la clasificación. Una página es juzgada relevante solo si es la página de entrada de algún sitio que está principalmente dedicado al tema de la consulta.

Dentro de LETOR Gov se encuentran los conjuntos TD2003 (*Top Distillation* 2003), TD2004 (*Top Distillation* 2004), NP2003 (*Named Page Finding* 2003), NP2004 (*Named Page Finding* 2004), HP2003 (*Homepage Finding* 2003) y HP2004 (*Homepage Finding* 2004). En esta colección se utilizan un total de 575 consultas y son considerados 64 rasgos para representar la correspondencia de cada par consulta - documento.

3.3 Medidas de evaluación

Las principales medidas de evaluación empleadas en R.I., se dividen en dos categorías que son concebidas de acuerdo a la relevancia:

- ✓ Relevancia binaria
  - Precisión en *n* (P@n: *Precision at n*)
  - Precisión media promedio (MAP: *Mean Average Precision*)

[5] Text REtrieval Conference, trec.nist.gov

- ✓ Múltiples niveles de relevancia
  - Ganancia cumulativa descontada normalizada (NDCG: *Normalized Discounted Cumulative Gain*)

3.3.1 P@n

La precisión en *n* mide la relevancia de los documentos que se encuentran por encima de *n* (incluyendo éste) como resultado del ordenamiento con respecto a una consulta dada:

$$P@n = \frac{\#\text{ de doc relevantes en los primeros } \boldsymbol{n} \text{ resultados}}{\boldsymbol{n}} \tag{3.1}$$

Por ejemplo, si los 10 primeros documentos retornados a partir de una consulta son (relevante, irrelevante, irrelevante, relevante, relevante, relevante, irrelevante, irrelevante, relevante, relevante), entonces de P@1 a P@10 los valores serán {1, 1/2, 1/3, 2/4, 3/5, 4/6, 4/7, 4/8, 5/9, 6/10} respectivamente. [17]

3.3.2 MAP

Para una única consulta, la precisión media se define como el promedio de los valores de P@n para todos los documentos relevantes:

$$AP = \frac{\sum_{n=1}^{N}(P@n \times rel(n))}{\#\ total\ de\ docs\ relevantes\ para\ esta\ consulta} \tag{3.2}$$

donde N es el número de documentos recuperados, y $rel(n)$ es una función binaria que retorna la relevancia del *n*-ésimo documento:

$$rel\ (n) = \begin{cases} 1 & : el\ n-\text{ésimo}\ documento\ es\ relevante \\ 0 & : el\ n-\text{ésimo}\ documento\ no\ es\ relevante \end{cases} \tag{3.3}$$

El cálculo de AP para el ejemplo citado en la explicación de P@n sería:

$$AP = \frac{1\times 1 + \frac{1}{2}\times 0 + \frac{1}{3}\times 0 + \frac{2}{4}\times 1 + \frac{3}{5}\times 1 + \frac{4}{6}\times 1 + \frac{4}{7}\times 0 + \frac{4}{8}\times 0 + \frac{5}{9}\times 1 + \frac{6}{10}\times 1}{6}$$

$$AP = \frac{1 + \frac{2}{4} + \frac{3}{5} + \frac{4}{6} + \frac{5}{9} + \frac{6}{10}}{6} = 0.6533$$

Dado un conjunto de consultas, el valor de MAP, será el promedio de los valores de AP para cada una de las consultas.

3.3.3 NDCG

Las medidas de evaluación P@n y MAP pueden manejar solamente casos con juicios de relevancia binarios: relevante o irrelevante. Para los casos en los que se quiere manipular múltiples niveles de juicios de relevancia surge en el año 2002 NDCG [18], que sigue dos reglas fundamentales:

1. Los documentos altamente relevantes son más valiosos que los documentos marginalmente relevantes.
2. Un documento (de cualquier nivel de relevancia) de baja posición en el ordenamiento, tiene menos valor para el usuario, porque su probabilidad de ser examinado por dicho usuario es mucho menor.

De acuerdo a las reglas anteriores, el valor NDCG de una lista de ordenamiento en la posición *n* es calculado como sigue:

$$N(n) = Z_n \sum_{j=1}^{n} \frac{2^{r(j)} - 1}{\log(1 + j)} \tag{3.4}$$

donde $r(j)$ es el valor de relevancia del *j*-ésimo documento en la lista de ordenamiento, y la constante de normalización Z_n es escogida para que la lista ideal obtenga un valor NDCG de 1. [17]

3.4 Comparación con métodos referenciados

Para la evaluación del desempeño del método propuesto y su comparación con los principales trabajos realizados en este campo investigativo; se siguieron las pautas experimentales publicadas en LETOR.

Se utilizó en el proceso de aprendizaje o entrenamiento para RankSupernova una validación cruzada con cinco particiones (*five-fold cross validation*), lo cual posibilita realizar comparaciones directas en términos de precisión con los métodos publicados para el Aprendizaje de la Ordenación.

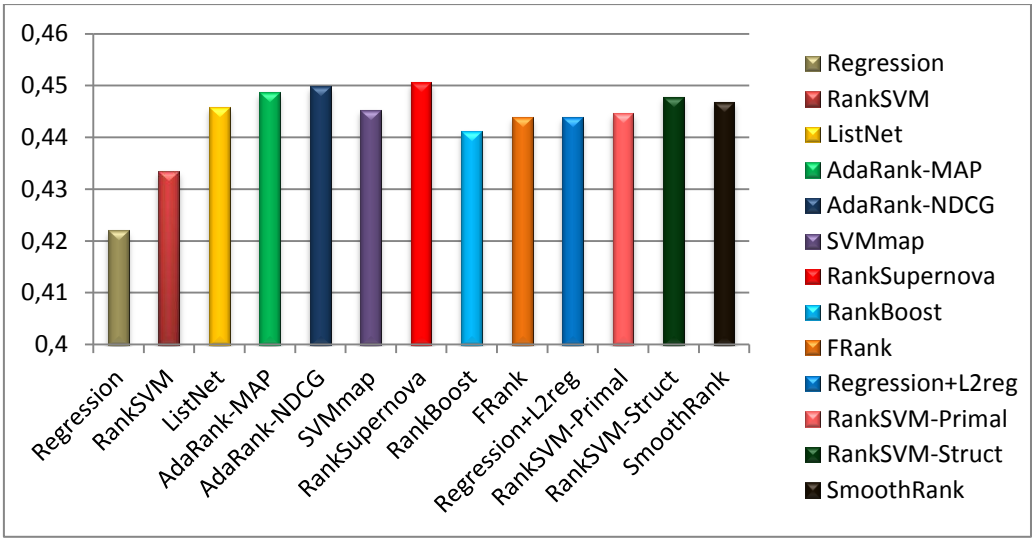
Los valores de los parámetros que forman parte de los datos de entrada del método y que fueron definidos en la sección 2.2.3, se fijaron tomando como

referencia las propuestas realizadas por la Ing. Eddy Mesa en su tesis de maestría [37], después de un estudio completo del comportamiento del método para diferentes tipos de funciones estándar, utilizadas en tareas de optimización. Para las constantes de impulso (F) y gravitación (G), empleadas para el cálculo del movimiento (ecuación 2.6) se tomó el valor 1. El radio de búsqueda o alcance de la explosión (λ) inicia con valor 20 (considerando el 50% de alcance en el área de búsqueda) y se le aplica un factor de reducción (∝) igual a 1,2.

La cantidad de partículas (*k*), movimientos (*T*) y reinicios (*RR*) se ajustaron de forma empírica según las mejoras que proporcionaban en los resultados. Se emplearon 25 partículas, 2 movimientos y los reinicios varían entre 3, 5, 10 y 20, en dependencia del conjunto de datos utilizado. En todos los experimentos llevados a cabo con RankSupernova, se consideró a MAP como función *E* en la expresión (2.10).

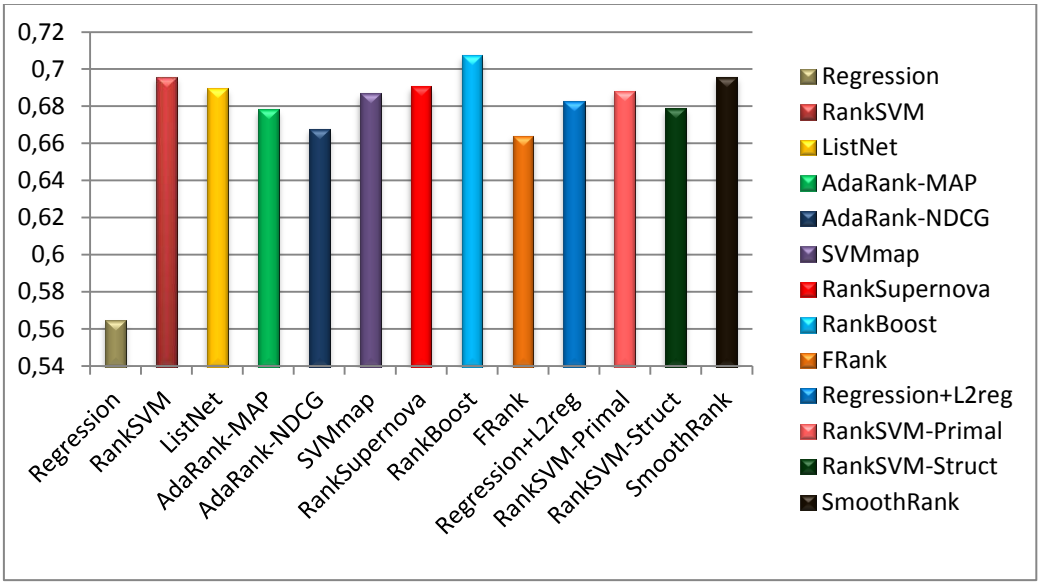
3.4.1 Desempeños obtenidos

En las Figuras 3.1 a 3.7 se muestran los resultados del rendimiento alcanzado en la ordenación sobre las colecciones de datos de LETOR 3.0 por cada algoritmo comparado; considerando la medida de desempeño MAP. Al lado de cada uno de los gráficos pueden apreciarse los valores obtenidos por cada algoritmo lo que confirma a primera vista el buen desempeño y estabilidad de RankSupernova.



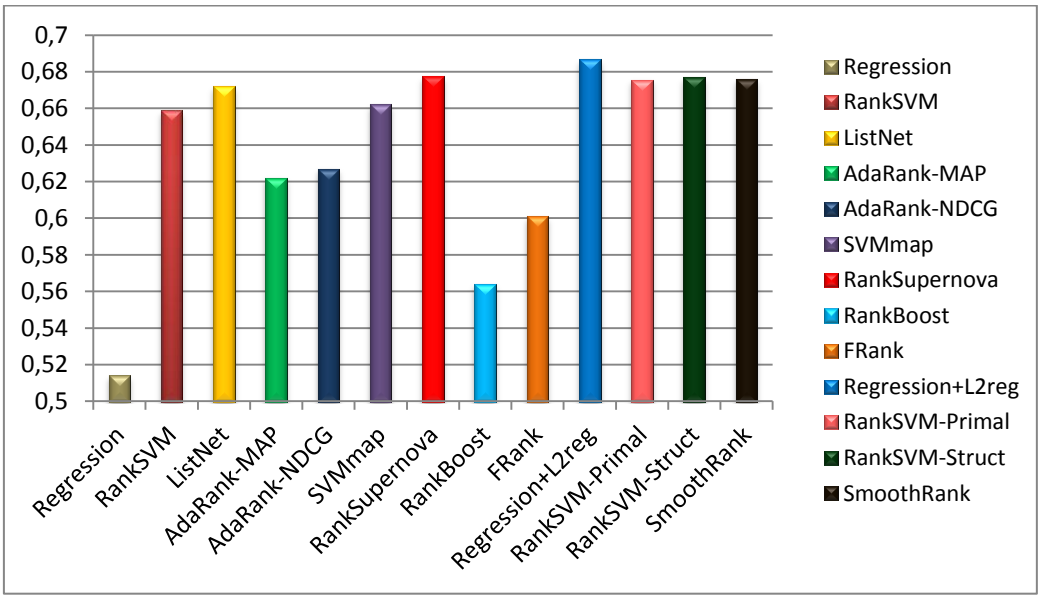
MAP
(-) 0,422
(-) 0,4334
(-) 0,4457
(-) 0,4487
(-) 0,4498
(-) 0,4453
0,4505 (1)
(-) 0,4411
(-) 0,4439
(-) 0,4439
(-) 0,4446
(-) 0,4478
(-) 0,4468

Figura 3.1: Rendimiento alcanzado en la ordenación sobre OHSUMED, considerando MAP como medida de desempeño.



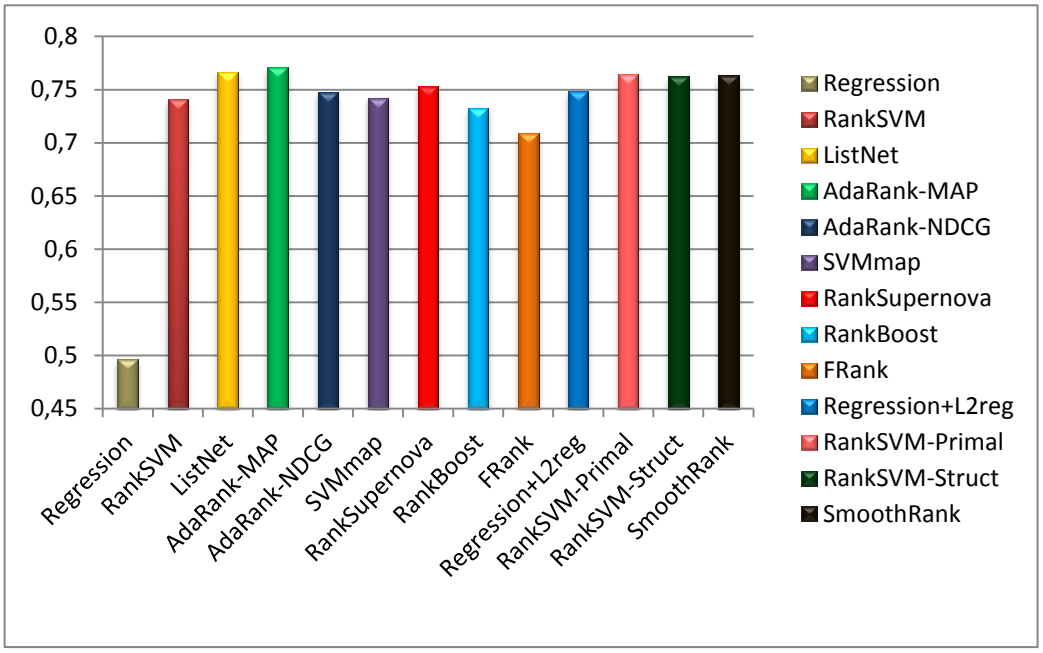
MAP
(-) 0,5644
(+) 0,6957
(-) 0,6895
(-) 0,6783
(-) 0,6678
(-) 0,6869
0,6906 (4)
(+) 0,7074
(-) 0,664
(-) 0,6824
(-) 0,6883
(-) 0,6788
(+) 0,6958

Figura 3.2: Rendimiento alcanzado en la ordenación sobre NP2003, considerando MAP como medida de desempeño.



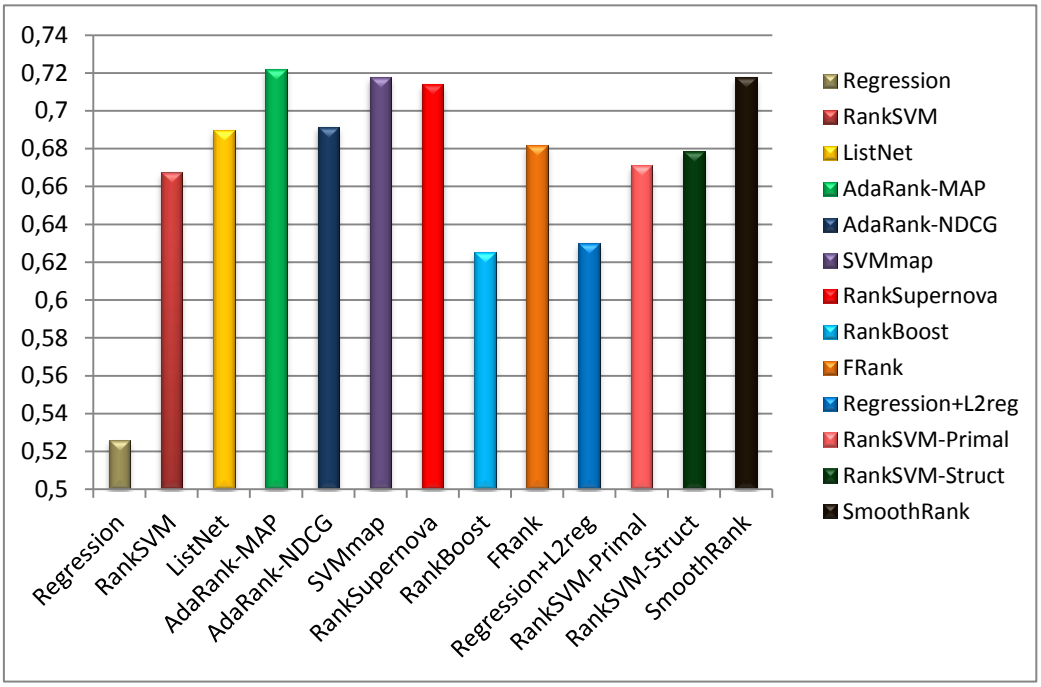
MAP
(-) 0,5142
(-) 0,6588
(-) 0,672
(-) 0,622
(-) 0,6269
(-) 0,662
0,6773 (2)
(-) 0,564
(-) 0,6008
(+) 0,6866
(-) 0,6755
(-) 0,6771
(-) 0,676

Figura 3.3: Rendimiento alcanzado en la ordenación sobre NP2004, considerando MAP como medida de desempeño.



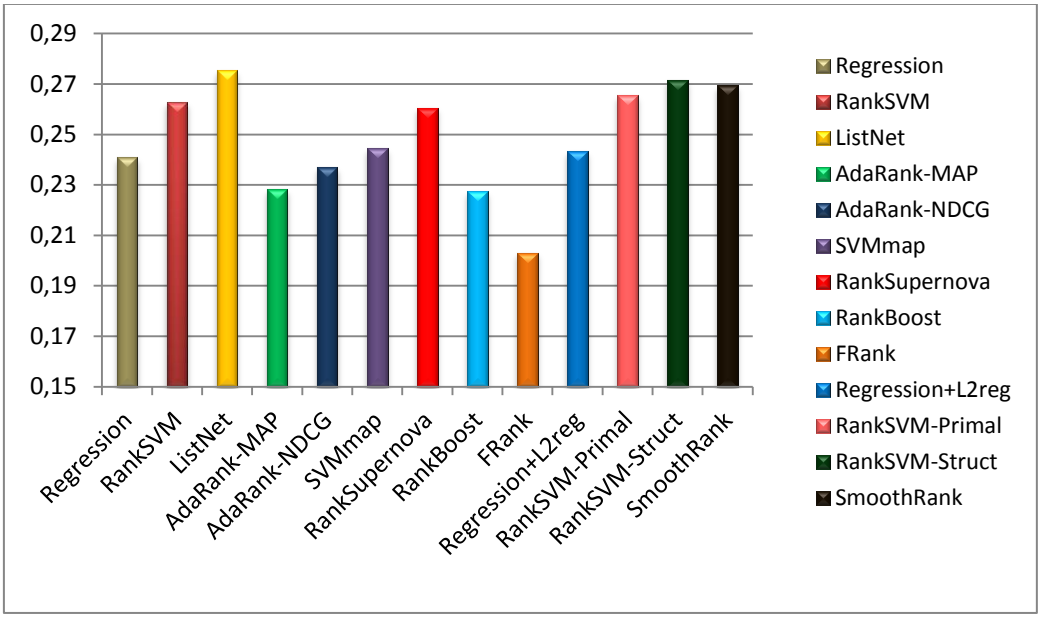
MAP
(-) 0,4968
(-) 0,7408
(+) 0,7659
(+) 0,771
(-) 0,748
(-) 0,7421
0,7526 (6)
(-) 0,733
(-) 0,7095
(-) 0,7486
(+) 0,7645
(+) 0,7625
(+) 0,7633

Figura 3.4: Rendimiento alcanzado en la ordenación sobre HP2003, considerando MAP como medida de desempeño.



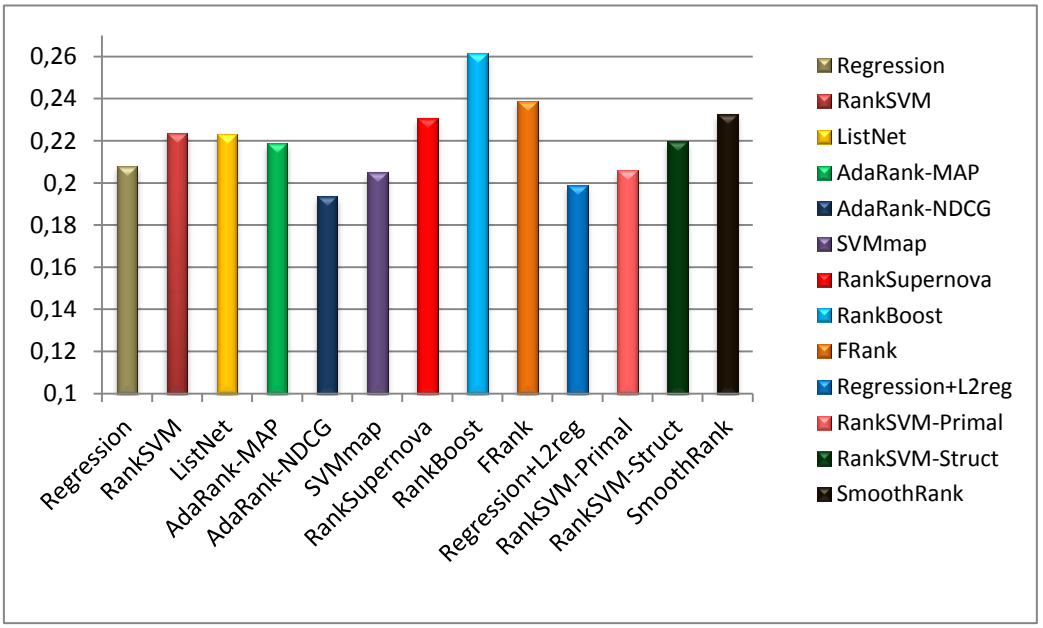
MAP
(-) 0,5256
(-) 0,6675
(-) 0,6899
(+) 0,7219
(-) 0,6914
(+) 0,7176
0,7138 (4)
(-) 0,6251
(-) 0,6817
(-) 0,6301
(-) 0,6712
(-) 0,6784
(+) 0,7174

Figura 3.5: Rendimiento alcanzado en la ordenación sobre HP2004, considerando MAP como medida de desempeño.



MAP
(-) 0,2409
(+) 0,2628
(+) 0,2753
(-) 0,2283
(-) 0,2368
(-) 0,2445
0,2605 (6)
(-) 0,2274
(-) 0,2031
(-) 0,2434
(+) 0,2653
(+) 0,2713
(+) 0,2695

Figura 3.6: Rendimiento alcanzado en la ordenación sobre TD2003, considerando MAP como medida de desempeño.



MAP
(-) 0,2078
(-) 0,2237
(-) 0,2231
(-) 0,2189
(-) 0,1936
(-) 0,2049
0,2307 (4)
(+) 0,2614
(+) 0,2388
(-) 0,1992
(-) 0,2061
(-) 0,2196
(+) 0,2326

Figura 3.7: Rendimiento alcanzado en la ordenación sobre TD2004, considerando MAP como medida de desempeño.

Con las partículas obtenidas en el aprendizaje empleando la medida de desempeño MAP como función *E*, se realiza la evaluación del método RankSupernova para las otras medidas de desempeño de la R.I.

En las tablas 3.1 a 3.7 pueden observarse los resultados obtenidos para P@n. Para facilitar el análisis de estas tablas, se coloca el signo (-) a aquellos métodos que no superan el resultado de RankSupernova, el signo (=) a los que tienen igual valor al método propuesto y el signo (+) a los que están por encima de dicho valor.

Al analizar estas tablas puede apreciarse cuan competitivos suelen ser los valores de rendimiento (precisión) obtenidos en las diez primeras posiciones de la ordenación por el método propuesto, en relación a los algoritmos publicados en LETOR.

Tabla 3.1: Valores del rendimiento alcanzado en la ordenación sobre OHSUMED, considerando P@n como medida de desempeño en las posiciones de la 1 a la 10.

(a)

Algoritmos	**P@1**	**P@2**	**P@3**	**P@4**	**P@5**
Regression	(-) 0,5965	(-) 0,6006	(-) 0,5768	(-) 0,5605	(-) 0,5337
RankSVM	(-) 0,5974	(-) 0,5494	(-) 0,5427	(-) 0,5443	(-) 0,5319
ListNet	(-) 0,6524	(-) 0,6093	(-) 0,6016	(-) 0,5745	(-) 0,5502
AdaRank-MAP	(-) 0,6338	(-) 0,5959	(-) 0,5895	(-) 0,5887	(-) 0,5674
AdaRank-NDCG	(-) 0,6719	(-) 0,6236	(-) 0,5984	(-) 0,5838	(-) 0,5767
SVMmap	(-) 0,6433	(-) 0,6197	(-) 0,5802	(-) 0,5768	(-) 0,5523
RankSupernova	0,6896	0,6615	0,617	0,5925	0,5821
RankBoost	(-) 0,5576	(-) 0,5481	(-) 0,5609	(-) 0,558	(-) 0,5447
FRank	(-) 0,6429	(-) 0,6195	(-) 0,5925	(-) 0,584	(-) 0,5638
Regression+L2reg	(-) 0,6437	(-) 0,6102	(-) 0,5896	(-) 0,5675	(-) 0,5505
RankSVM-Primal	(-) 0,6156	(-) 0,6052	(-) 0,5866	(-) 0,5773	(-) 0,5639
RankSVM-Struct	(-) 0,6338	(-) 0,6097	(-) 0,5898	(-) 0,5915	(-) 0,5696
SmoothRank	(-) 0,671	(-) 0,6377	(+) 0,6172	(-) 0,586	(-) 0,5711

(b)

Algoritmos	**P@6**	**P@7**	**P@8**	**P@9**	**P@10**
Regression	(-) 0,5045	(-) 0,5001	(-) 0,4837	(-) 0,4752	(-) 0,4666
RankSVM	(-) 0,5253	(-) 0,5097	(-) 0,4933	(-) 0,492	(-) 0,4864
ListNet	(-) 0,5373	(-) 0,5267	(-) 0,5236	(-) 0,5138	(-) 0,4975
AdaRank-MAP	(-) 0,5566	(-) 0,5392	(-) 0,5239	(-) 0,5077	(-) 0,4976
AdaRank-NDCG	(-) 0,5563	(+) 0,5511	(-) 0,5354	(-) 0,5211	(+) 0,5087
SVMmap	(-) 0,5281	(-) 0,5108	(-) 0,506	(-) 0,4928	(-) 0,491

RankSupernova	0,5607	0,5429	0,5356	0,5213	0,5031
RankBoost	(-) 0,5297	(-) 0,5241	(-) 0,513	(-) 0,5024	(-) 0,4966
FRank	(-) 0,5517	(+) 0,5445	(-) 0,5251	(-) 0,5152	(-) 0,5016
Regression+L2reg	(-) 0,5423	(-) 0,5189	(-) 0,5144	(-) 0,5056	(-) 0,5014
RankSVM-Primal	(-) 0,5364	(-) 0,5286	(-) 0,5192	(-) 0,5087	(+) 0,5071
RankSVM-Struct	(-) 0,5426	(-) 0,5325	(-) 0,5249	(-) 0,5203	(+) 0,5071
SmoothRank	(-) 0,5531	(-) 0,5376	(-) 0,5261	(-) 0,5109	(+) 0,506

Tabla 3.2: Valores del rendimiento alcanzado en la ordenación sobre NP2003, considerando P@n como medida de desempeño en las posiciones de la 1 a la 10.

(a)

Algoritmos	**P@1**	**P@2**	**P@3**	**P@4**	**P@5**
Regression	(-) 0,4467	(-) 0,2833	(-) 0,22	(-) 0,1767	(-) 0,1453
RankSVM	(-) 0,58	(+) 0,37	(+) 0,2711	(+) 0,2083	(+) 0,1707
ListNet	(-) 0,5667	(+) 0,37	(+) 0,2667	(+) 0,2083	(+) 0,172
AdaRank-MAP	(-) 0,58	(+) 0,3567	(-) 0,2511	(-) 0,1917	(-) 0,16
AdaRank-NDCG	(-) 0,56	(-) 0,3467	(-) 0,2467	(-) 0,1967	(-) 0,1613
SVMmap	(-) 0,56	(+) 0,38	(+) 0,2689	(+) 0,21	(+) 0,1707
RankSupernova	0,5867	0,35	0,26	0,2067	0,1693
RankBoost	(+) 0,6	(+) 0,3767	(+) 0,2689	(-) 0,205	(=) 0,1693
FRank	(-) 0,54	(+) 0,3533	(-) 0,2533	(-) 0,2033	(-) 0,168
Regression+L2reg	(-) 0,5467	(+) 0,37	(+) 0,2733	(+) 0,2117	(+) 0,172
RankSVM-Primal	(-) 0,5733	(+) 0,36	(+) 0,2733	(+) 0,2117	(=) 0,1693
RankSVM-Struct	(-) 0,5533	(+) 0,3667	(+) 0,2622	(+) 0,2083	(+) 0,1707
SmoothRank	(-) 0,58	(+) 0,3733	(=) 0,26	(+) 0,2083	(+) 0,172

(b)

Algoritmos	**P@6**	**P@7**	**P@8**	**P@9**	**P@10**
Regression	(-) 0,1256	(-) 0,1105	(-) 0,0983	(-) 0,0889	(-) 0,0813
RankSVM	(-) 0,1433	(-) 0,1267	(+) 0,1117	(+) 0,1	(+) 0,092
ListNet	(=) 0,1456	(+) 0,1295	(+) 0,1133	(+) 0,1015	(+) 0,0927
AdaRank-MAP	(-) 0,1367	(-) 0,1181	(+) 0,105	(+) 0,0941	(+) 0,0867
AdaRank-NDCG	(-) 0,14	(-) 0,1229	(+) 0,1083	(+) 0,0985	(+) 0,0893
SVMmap	(-) 0,1433	(-) 0,1257	(+) 0,11	(+) 0,0985	(+) 0,0893
RankSupernova	0,1456	0,1276	0,1033	0,0933	0,0847
RankBoost	(-) 0,1433	(=) 0,1276	(+) 0,1125	(+) 0,103	(+) 0,094
FRank	(-) 0,1433	(-) 0,1238	(+) 0,11	(+) 0,0978	(+) 0,0907
Regression+L2reg	(=) 0,1456	(-) 0,1267	(+) 0,1117	(+) 0,1	(+) 0,0913
RankSVM-Primal	(-) 0,1422	(-) 0,1257	(+) 0,11	(+) 0,0993	(+) 0,0907
RankSVM-Struct	(-) 0,1433	(-) 0,1257	(+) 0,1117	(+) 0,1	(+) 0,092
SmoothRank	(=) 0,1456	(-) 0,1257	(+) 0,11	(+) 0,1	(+) 0,09

Tabla 3.3: Valores del rendimiento alcanzado en la ordenación sobre NP2004, considerando P@n como medida de desempeño en las posiciones de la 1 a la 10.

(a)

Algoritmos	P@1	P@2	P@3	P@4	P@5
Regression	(-) 0,3733	(-) 0,2667	(-) 0,2	(-) 0,1733	(-) 0,144
RankSVM	(-) 0,5067	(=) 0,3733	(-) 0,2622	(+) 0,22	(=) 0,1787
ListNet	(=) 0,5333	(=) 0,3733	(=) 0,2667	(+) 0,2167	(=) 0,1787
AdaRank-MAP	(-) 0,48	(-) 0,3333	(-) 0,2444	(-) 0,1933	(-) 0,1627
AdaRank-NDCG	(-) 0,5067	(-) 0,32	(-) 0,2489	(-) 0,2067	(-) 0,1653
SVMmap	(-) 0,52	(-) 0,36	(=) 0,2667	(+) 0,22	(=) 0,1787
RankSupernova	0,5333	0,3733	0,2667	0,2133	0,1787
RankBoost	(-) 0,4267	(-) 0,2867	(-) 0,2311	(-) 0,1833	(-) 0,152
FRank	(-) 0,48	(-) 0,3133	(-) 0,2356	(-) 0,19	(-) 0,16
Regression+L2reg	(+) 0,5733	(-) 0,36	(-) 0,2622	(+) 0,2167	(-) 0,176
RankSVM-Primal	(+) 0,56	(-) 0,36	(-) 0,2533	(=) 0,2133	(-) 0,1733
RankSVM-Struct	(+) 0,56	(-) 0,36	(-) 0,2578	(+) 0,2167	(-) 0,1733
SmoothRank	(+) 0,5467	(-) 0,36	(=) 0,2667	(+) 0,22	(=) 0,1787

(b)

Algoritmos	P@6	P@7	P@8	P@9	P@10
Regression	(-) 0,1311	(-) 0,1181	(-) 0,1033	(-) 0,0919	(-) 0,0827
RankSVM	(-) 0,1489	(-) 0,1295	(-) 0,1133	(-) 0,1022	(-) 0,0933
ListNet	(+) 0,1533	(=) 0,1333	(=) 0,1183	(=) 0,1052	(=) 0,0947
AdaRank-MAP	(-) 0,14	(-) 0,1219	(-) 0,11	(-) 0,0978	(-) 0,088
AdaRank-NDCG	(-) 0,1422	(-) 0,1238	(-) 0,1133	(-) 0,1007	(-) 0,0907
SVMmap	(+) 0,1533	(=) 0,1333	(=) 0,1183	(=) 0,1052	(+) 0,096
RankSupernova	0,1511	0,1333	0,1183	0,1052	0,0947
RankBoost	(-) 0,1333	(-) 0,12	(-) 0,1067	(-) 0,0948	(-) 0,088
FRank	(-) 0,14	(-) 0,1238	(-) 0,11	(-) 0,1007	(-) 0,0933
Regression+L2reg	(+) 0,1533	(+) 0,1352	(=) 0,1183	(+) 0,1067	(+) 0,096
RankSVM-Primal	(=) 0,1511	(-) 0,1314	(-) 0,1167	(=) 0,1052	(=) 0,0947
RankSVM-Struct	(=) 0,1511	(-) 0,1314	(-) 0,1167	(=) 0,1052	(=) 0,0947
SmoothRank	(+) 0,1556	(+) 0,1371	(+) 0,12	(+) 0,1067	(+) 0,096

Tabla 3.4: Valores del rendimiento alcanzado en la ordenación sobre HP2003, considerando P@n como medida de desempeño en las posiciones de la 1 a la 10.

(a)

Algoritmos	P@1	P@2	P@3	P@4	P@5
Regression	(-) 0,42	(-) 0,27	(-) 0,2111	(-) 0,1733	(-) 0,1453
RankSVM	(-) 0,6933	(-) 0,4233	(+) 0,3089	(+) 0,24	(+) 0,1987
ListNet	(+) 0,72	(+) 0,4533	(+) 0,32	(+) 0,2483	(+) 0,204

AdaRank-MAP	(+) 0,7333	(+) 0,45	(+) 0,3089	(+) 0,24	(+) 0,1987
AdaRank-NDCG	(=) 0,7133	(+) 0,4367	(+) 0,3111	(-) 0,2367	(=) 0,1947
SVMmap	(=) 0,7133	(+) 0,4333	(+) 0,3089	(+) 0,24	(=) 0,1947
RankSupernova	0,7133	0,43	0,3044	0,2383	0,1947
RankBoost	(-) 0,6667	(+) 0,4333	(+) 0,3111	(-) 0,2367	(+) 0,1987
FRank	(-) 0,6533	(-) 0,4133	(-) 0,2889	(-) 0,2333	(+) 0,1987
Regression+L2reg	(-) 0,6933	(+) 0,4467	(+) 0,32	(+) 0,2467	(+) 0,1973
RankSVM-Primal	(+) 0,74	(+) 0,4333	(+) 0,3133	(+) 0,2433	(+) 0,2
RankSVM-Struct	(+) 0,74	(+) 0,4333	(+) 0,3089	(+) 0,24	(+) 0,1987
SmoothRank	(=) 0,7133	(+) 0,4467	(+) 0,3244	(+) 0,2517	(+) 0,2067

(b)

Algoritmos	**P@6**	**P@7**	**P@8**	**P@9**	**P@10**
Regression	(-) 0,1233	(-) 0,1143	(-) 0,1008	(-) 0,0926	(-) 0,0887
RankSVM	(+) 0,1689	(-) 0,1467	(-) 0,1292	(-) 0,1148	(-) 0,104
ListNet	(+) 0,1722	(+) 0,1495	(+) 0,1308	(=) 0,117	(+) 0,1067
AdaRank-MAP	(+) 0,1689	(=) 0,1476	(=) 0,13	(-) 0,1163	(=) 0,106
AdaRank-NDCG	(=) 0,1644	(-) 0,1419	(-) 0,1242	(-) 0,1104	(-) 0,1
SVMmap	(+) 0,1656	(-) 0,1429	(-) 0,125	(-) 0,1111	(-) 0,1
RankSupernova	0,1644	0,1476	0,13	0,117	0,106
RankBoost	(+) 0,1678	(=) 0,1476	(=) 0,13	(-) 0,1163	(-) 0,1053
FRank	(+) 0,17	(-) 0,1467	(-) 0,1283	(-) 0,1156	(+) 0,1067
Regression+L2reg	(+) 0,1656	(-) 0,1419	(-) 0,125	(-) 0,1111	(-) 0,1027
RankSVM-Primal	(+) 0,1689	(-) 0,1467	(-) 0,1283	(-) 0,1148	(-) 0,104
RankSVM-Struct	(+) 0,1667	(-) 0,1438	(-) 0,1283	(-) 0,1141	(-) 0,1027
SmoothRank	(+) 0,1733	(+) 0,1486	(=) 0,13	(-) 0,1156	(-) 0,1053

Tabla 3.5: Valores del rendimiento alcanzado en la ordenación sobre HP2004, considerando P@n como medida de desempeño en las posiciones de la 1 a la 10.

(a)

Algoritmos	**P@1**	**P@2**	**P@3**	**P@4**	**P@5**
Regression	(-) 0,3867	(-) 0,2867	(-) 0,2133	(-) 0,1767	(-) 0,144
RankSVM	(-) 0,5733	(-) 0,3667	(-) 0,2667	(-) 0,2167	(-) 0,1787
ListNet	(-) 0,6	(-) 0,3733	(-) 0,2711	(+) 0,23	(=) 0,1893
AdaRank-MAP	(-) 0,6133	(+) 0,4067	(+) 0,2978	(+) 0,23	(-) 0,1867
AdaRank-NDCG	(-) 0,5867	(-) 0,3867	(-) 0,2711	(=) 0,2233	(-) 0,1813
SVMmap	(-) 0,6267	(=) 0,4	(=) 0,28	(+) 0,2333	(=) 0,1893
RankSupernova	0,64	0,4	0,28	0,2233	0,1893
RankBoost	(-) 0,5067	(-) 0,3467	(-) 0,2533	(-) 0,1933	(-) 0,1653
FRank	(-) 0,6	(-) 0,3733	(-) 0,2622	(-) 0,2033	(-) 0,168

Regression+L2reg	(-) 0,5333	(-) 0,34	(-) 0,2489	(-) 0,2033	(-) 0,168
RankSVM-Primal	(-) 0,5733	(-) 0,3733	(-) 0,2667	(-) 0,2167	(-) 0,1787
RankSVM-Struct	(-) 0,5867	(-) 0,3667	(-) 0,2756	(-) 0,22	(-) 0,1787
SmoothRank	(-) 0,6133	(=) 0,4	(+) 0,2978	(+) 0,23	(+) 0,1947

(b)

Algoritmos	P@6	P@7	P@8	P@9	P@10
Regression	(-) 0,1267	(-) 0,1086	(-) 0,0983	(-) 0,0919	(-) 0,084
RankSVM	(-) 0,1533	(-) 0,1314	(-) 0,1167	(-) 0,1052	(-) 0,096
ListNet	(=) 0,1622	(-) 0,141	(+) 0,1233	(=) 0,1096	(=) 0,0987
AdaRank-MAP	(-) 0,1578	(-) 0,1352	(-) 0,1183	(-) 0,1052	(-) 0,0947
AdaRank-NDCG	(-) 0,1533	(-) 0,1314	(-) 0,1167	(-) 0,1052	(-) 0,096
SVMmap	(-) 0,16	(-) 0,1371	(-) 0,12	(-) 0,1067	(-) 0,096
RankSupernova	0,1622	0,139	0,1217	0,1096	0,0987
RankBoost	(-) 0,14	(-) 0,12	(-) 0,11	(-) 0,1022	(-) 0,092
FRank	(-) 0,1444	(-) 0,1238	(-) 0,1083	(-) 0,0978	(-) 0,0893
Regression+L2reg	(-) 0,1422	(-) 0,1257	(-) 0,11	(-) 0,1007	(-) 0,0907
RankSVM-Primal	(-) 0,1556	(-) 0,1352	(-) 0,1183	(-) 0,1067	(-) 0,096
RankSVM-Struct	(-) 0,1556	(-) 0,1333	(-) 0,1183	(-) 0,1052	(-) 0,0947
SmoothRank	(+) 0,1644	(+) 0,141	(+) 0,1233	(=) 0,1096	(=) 0,0987

Tabla 3.6: Valores del rendimiento alcanzado en la ordenación sobre TD2003, considerando P@n como medida de desempeño en las posiciones de la 1 a la 10.

(a)

Algoritmos	P@1	P@2	P@3	P@4	P@5
Regression	(-) 0,32	(-) 0,3	(-) 0,26	(-) 0,245	(-) 0,216
RankSVM	(-) 0,32	(-) 0,31	(-) 0,2933	(+) 0,285	(+) 0,276
ListNet	(=) 0,4	(=) 0,33	(-) 0,2933	(-) 0,255	(+) 0,252
AdaRank-MAP	(-) 0,26	(-) 0,29	(-) 0,26	(-) 0,225	(-) 0,208
AdaRank-NDCG	(-) 0,36	(-) 0,27	(-) 0,24	(-) 0,225	(-) 0,204
SVMmap	(-) 0,32	(-) 0,31	(-) 0,2533	(=) 0,26	(-) 0,232
RankSupernova	0,4	0,33	0,3	0,26	0,248
RankBoost	(-) 0,28	(-) 0,27	(-) 0,28	(-) 0,255	(-) 0,232
FRank	(-) 0,3	(-) 0,27	(-) 0,2333	(-) 0,2	(-) 0,172
Regression+L2reg	(-) 0,34	(+) 0,34	(=) 0,3	(-) 0,255	(-) 0,236
RankSVM-Primal	(-) 0,32	(+) 0,35	(-) 0,2933	(+) 0,285	(+) 0,272
RankSVM-Struct	(-) 0,34	(-) 0,32	(-) 0,2867	(+) 0,285	(+) 0,276
SmoothRank	(-) 0,38	(+) 0,34	(-) 0,2733	(=) 0,26	(-) 0,244

(b)

Algoritmos	P@6	P@7	P@8	P@9	P@10
Regression	(-) 0,2133	(-) 0,2057	(+) 0,195	(+) 0,1844	(-) 0,178
RankSVM	(+) 0,25	(+) 0,2229	(+) 0,2075	(+) 0,2	(+) 0,188
ListNet	(+) 0,2533	(+) 0,2486	(+) 0,2225	(+) 0,2133	(+) 0,2
AdaRank-MAP	(-) 0,2	(-) 0,1943	(-) 0,18	(-) 0,1667	(-) 0,158
AdaRank-NDCG	(-) 0,1933	(-) 0,1971	(-) 0,1825	(-) 0,1733	(-) 0,164
SVMmap	(-) 0,21	(-) 0,1971	(-) 0,1825	(-) 0,1733	(-) 0,17
RankSupernova	0,2333	0,2086	0,185	0,1822	0,18
RankBoost	(-) 0,22	(-) 0,2	(+) 0,1875	(-) 0,1778	(-) 0,17
FRank	(-) 0,1733	(-) 0,1657	(-) 0,1575	(-) 0,1533	(-) 0,152
Regression+L2reg	(-) 0,2067	(=) 0,2086	(+) 0,1975	(=) 0,1822	(-) 0,176
RankSVM-Primal	(+) 0,2533	(+) 0,2314	(+) 0,21	(+) 0,2044	(+) 0,194
RankSVM-Struct	(+) 0,25	(+) 0,2343	(+) 0,2075	(+) 0,2	(+) 0,186
SmoothRank	(-) 0,2267	(+) 0,2229	(+) 0,21	(+) 0,1956	(+) 0,188

Tabla 3.7: Valores del rendimiento alcanzado en la ordenación sobre TD2004, considerando P@n como medida de desempeño en las posiciones de la 1 a la 10.

(a)

Algoritmos	P@1	P@2	P@3	P@4	P@5
Regression	(-) 0,36	(-) 0,34	(-) 0,3333	(-) 0,32	(-) 0,312
RankSVM	(-) 0,4133	(-) 0,3467	(-) 0,3467	(-) 0,3333	(-) 0,3013
ListNet	(-) 0,36	(-) 0,3467	(-) 0,36	(-) 0,3367	(-) 0,3067
AdaRank-MAP	(-) 0,4133	(=) 0,3933	(+) 0,3689	(+) 0,3533	(-) 0,328
AdaRank-NDCG	(=) 0,4267	(-) 0,38	(=) 0,3644	(-) 0,3333	(-) 0,328
SVMmap	(-) 0,2933	(-) 0,3067	(-) 0,3022	(-) 0,2933	(-) 0,2907
RankSupernova	0,4267	0,3933	0,3644	0,35	0,336
RankBoost	(+) 0,5067	(+) 0,4333	(+) 0,4267	(+) 0,3833	(+) 0,352
FRank	(+) 0,4933	(+) 0,4067	(+) 0,3778	(-) 0,3267	(-) 0,3333
Regression+L2reg	(-) 0,2933	(-) 0,32	(-) 0,2978	(-) 0,2833	(-) 0,272
RankSVM-Primal	(-) 0,3067	(-) 0,3067	(-) 0,3156	(-) 0,2933	(-) 0,2933
RankSVM-Struct	(-) 0,3467	(-) 0,3467	(-) 0,3333	(-) 0,3167	(-) 0,296
SmoothRank	(-) 0,4	(+) 0,42	(+) 0,3689	(-) 0,3467	(-) 0,32

(b)

Algoritmos	P@6	P@7	P@8	P@9	P@10
Regression	(-) 0,2867	(-) 0,2705	(-) 0,2683	(-) 0,2593	(-) 0,2493
RankSVM	(-) 0,2867	(-) 0,28	(-) 0,2667	(-) 0,2548	(-) 0,252
ListNet	(-) 0,2933	(-) 0,2857	(-) 0,2717	(-) 0,2637	(-) 0,256
AdaRank-MAP	(-) 0,3111	(-) 0,2914	(-) 0,2733	(-) 0,2637	(-) 0,2493

AdaRank-NDCG	(-) 0,3	(-) 0,2838	(-) 0,2783	(-) 0,2622	(-) 0,248
SVMmap	(-) 0,2756	(-) 0,2743	(-) 0,2617	(-) 0,2519	(-) 0,2467
RankSupernova	0,3244	0,299	0,2867	0,28	0,264
RankBoost	(+) 0,3311	(+) 0,3086	(+) 0,2983	(+) 0,283	(+) 0,2747
FRank	(-) 0,3156	(=) 0,299	(-) 0,2817	(-) 0,2681	(-) 0,2627
Regression+L2reg	(-) 0,2644	(-) 0,2629	(-) 0,25	(-) 0,2415	(-) 0,2347
RankSVM-Primal	(-) 0,2778	(-) 0,2648	(-) 0,255	(-) 0,2533	(-) 0,2427
RankSVM-Struct	(-) 0,2822	(-) 0,2724	(-) 0,275	(-) 0,2593	(-) 0,256
SmoothRank	(-) 0,2956	(-) 0,2933	(+) 0,2883	(-) 0,2785	(+) 0,2653

Al igual que se hizo para la medida de desempeño P@n, en las tablas 3.8 a 3.14 se muestran los resultados de la evaluación de RankSupernova para NDCG.

Tabla 3.8: Valores del rendimiento alcanzado en la ordenación sobre OHSUMED, considerando NDCG como medida de desempeño en las posiciones de la 1 a la 10.

(a)

Algoritmos	NDCG@1	NDCG@2	NDCG@3	NDCG@4	NDCG@5
Regression	(-) 0,4456	(-) 0,4532	(-) 0,4426	(-) 0,4368	(-) 0,4278
RankSVM	(-) 0,4958	(-) 0,4331	(-) 0,4207	(-) 0,424	(-) 0,4164
ListNet	(-) 0,5326	(-) 0,481	(-) 0,4732	(-) 0,4561	(-) 0,4432
AdaRank-MAP	(-) 0,5388	(-) 0,4789	(-) 0,4682	(-) 0,4721	(-) 0,4613
AdaRank-NDCG	(-) 0,533	(-) 0,4922	(-) 0,479	(-) 0,4688	(-) 0,4673
SVMmap	(-) 0,5229	(-) 0,4909	(-) 0,4663	(-) 0,4625	(-) 0,4516
RankSupernova	0,5823	0,5323	0,4953	0,4777	0,4773
RankBoost	(-) 0,4632	(-) 0,4504	(-) 0,4555	(-) 0,4543	(-) 0,4494
FRank	(-) 0,53	(-) 0,5008	(-) 0,4812	(-) 0,4694	(-) 0,4588
Regression+L2reg	(-) 0,5361	(-) 0,4784	(-) 0,474	(-) 0,4601	(-) 0,4568
RankSVM-Primal	(-) 0,546	(-) 0,501	(-) 0,4855	(-) 0,4766	(-) 0,4689
RankSVM-Struct	(-) 0,5515	(-) 0,5	(-) 0,485	(+) 0,482	(-) 0,4729
SmoothRank	(-) 0,5576	(-) 0,5149	(+) 0,4964	(+) 0,485	(+) 0,4776

(b)

Algoritmos	NDCG@6	NDCG@7	NDCG@8	NDCG@9	NDCG@10
Regression	(-) 0,4216	(-) 0,4217	(-) 0,4187	(-) 0,4136	(-) 0,411
RankSVM	(-) 0,4159	(-) 0,4133	(-) 0,4072	(-) 0,4124	(-) 0,414
ListNet	(-) 0,44	(-) 0,4409	(-) 0,446	(-) 0,4459	(-) 0,441
AdaRank-MAP	(-) 0,4579	(-) 0,4558	(-) 0,4506	(-) 0,4464	(-) 0,4429
AdaRank-NDCG	(-) 0,4597	(-) 0,4596	(-) 0,4575	(-) 0,4541	(-) 0,4496
SVMmap	(-) 0,4411	(-) 0,4335	(-) 0,4332	(-) 0,4304	(-) 0,4319
RankSupernova	0,4701	0,4649	0,4644	0,4613	0,4538

RankBoost	(-) 0,4439	(-) 0,4412	(-) 0,4361	(-) 0,4326	(-) 0,4302
FRank	(-) 0,455	(-) 0,4529	(-) 0,4476	(-) 0,446	(-) 0,4433
Regression+L2reg	(-) 0,4536	(-) 0,4449	(-) 0,4444	(-) 0,4427	(-) 0,4436
RankSVM-Primal	(-) 0,4552	(-) 0,4534	(-) 0,45	(-) 0,449	(-) 0,4504
RankSVM-Struct	(-) 0,4584	(-) 0,457	(-) 0,4587	(-) 0,4568	(-) 0,4523
SmoothRank	(+) 0,4702	(+) 0,4667	(-) 0,4608	(-) 0,4555	(+) 0,4568

Tabla 3.9: Valores del rendimiento alcanzado en la ordenación sobre NP2003, considerando NDCG como medida de desempeño en las posiciones de la 1 a la 10.

(a)

Algoritmos	NDCG@1	NDCG@2	NDCG@3	NDCG@4	NDCG@5
Regression	(-) 0,4467	(-) 0,5567	(-) 0,6135	(-) 0,6351	(-) 0,6423
RankSVM	(-) 0,58	(+) 0,7233	(+) 0,7654	(+) 0,7737	(+) 0,7823
ListNet	(-) 0,5667	(+) 0,72	(+) 0,7579	(+) 0,7729	(+) 0,7843
AdaRank-MAP	(-) 0,58	(+) 0,7033	(-) 0,7286	(-) 0,7352	(-) 0,7482
AdaRank-NDCG	(-) 0,56	(-) 0,6867	(-) 0,7161	(-) 0,7361	(-) 0,7447
SVMmap	(-) 0,56	(+) 0,74	(+) 0,7673	(+) 0,7823	(+) 0,7881
RankSupernova	0,5867	0,6933	0,7375	0,7592	0,7678
RankBoost	(+) 0,6	(+) 0,73	(+) 0,7636	(+) 0,7703	(+) 0,7818
FRank	(-) 0,54	(+) 0,6967	(-) 0,7261	(-) 0,7494	(-) 0,7595
Regression+L2reg	(-) 0,5467	(+) 0,7267	(+) 0,7729	(+) 0,7829	(+) 0,7887
RankSVM-Primal	(-) 0,5733	(+) 0,7	(+) 0,7631	(+) 0,7748	(+) 0,7748
RankSVM-Struct	(-) 0,5533	(+) 0,7167	(+) 0,7503	(+) 0,7703	(+) 0,7789
SmoothRank	(-) 0,58	(+) 0,7333	(+) 0,7544	(+) 0,7777	(+) 0,7892

(b)

Algoritmos	NDCG@6	NDCG@7	NDCG@8	NDCG@9	NDCG@10
Regression	(-) 0,6526	(-) 0,6597	(-) 0,6597	(-) 0,6629	(-) 0,6659
RankSVM	(+) 0,7849	(+) 0,7944	(+) 0,7922	(+) 0,7943	(+) 0,8003
ListNet	(+) 0,7882	(+) 0,8001	(+) 0,7956	(+) 0,7977	(+) 0,8018
AdaRank-MAP	(-) 0,7546	(-) 0,757	(+) 0,757	(+) 0,7591	(+) 0,7641
AdaRank-NDCG	(-) 0,7563	(-) 0,7623	(+) 0,7589	(+) 0,7652	(+) 0,7672
SVMmap	(+) 0,7907	(+) 0,7978	(+) 0,7933	(+) 0,7954	(+) 0,7975
RankSupernova	0,7768	0,7839	0,7222	0,7264	0,7284
RankBoost	(+) 0,787	(+) 0,7976	(+) 0,7954	(+) 0,8028	(+) 0,8068
FRank	(-) 0,7659	(-) 0,7683	(+) 0,7683	(+) 0,7683	(+) 0,7763
Regression+L2reg	(+) 0,7938	(+) 0,7986	(+) 0,7964	(+) 0,7985	(+) 0,8025
RankSVM-Primal	(+) 0,7773	(+) 0,7857	(+) 0,7812	(+) 0,7854	(+) 0,7894
RankSVM-Struct	(+) 0,7802	(+) 0,7873	(+) 0,7873	(+) 0,7894	(+) 0,7955
SmoothRank	(+) 0,7943	(+) 0,7967	(+) 0,7923	(+) 0,7986	(+) 0,7986

Tabla 3.10: Valores del rendimiento alcanzado en la ordenación sobre NP2004, considerando NDCG como medida de desempeño en las posiciones de la 1 a la 10.

(a)

Algoritmos	NDCG@1	NDCG@2	NDCG@3	NDCG@4	NDCG@5
Regression	(-) 0,3733	(-) 0,5133	(-) 0,5554	(-) 0,6021	(-) 0,6135
RankSVM	(-) 0,5067	(=) 0,7267	(-) 0,7503	(+) 0,7928	(-) 0,7957
ListNet	(=) 0,5333	(=) 0,7267	(=) 0,7587	(+) 0,7879	(-) 0,7965
AdaRank-MAP	(-) 0,48	(-) 0,66	(-) 0,6979	(-) 0,7137	(-) 0,731
AdaRank-NDCG	(-) 0,5067	(-) 0,6133	(-) 0,6722	(-) 0,7122	(-) 0,7122
SVMmap	(-) 0,52	(-) 0,7	(-) 0,7489	(+) 0,7847	(-) 0,7869
RankSupernova	0,5333	0,7267	0,7587	0,7813	0,7985
RankBoost	(-) 0,4267	(-) 0,5533	(-) 0,6274	(-) 0,6433	(-) 0,6512
FRank	(-) 0,48	(-) 0,6	(-) 0,6431	(-) 0,6697	(-) 0,687
Regression+L2reg	(+) 0,5733	(-) 0,7	(-) 0,7352	(-) 0,7752	(-) 0,7774
RankSVM-Primal	(+) 0,56	(-) 0,7	(-) 0,7236	(-) 0,7662	(-) 0,7719
RankSVM-Struct	(+) 0,56	(-) 0,7	(-) 0,7321	(-) 0,7746	(-) 0,7746
SmoothRank	(+) 0,5467	(-) 0,7	(-) 0,7437	(-) 0,7803	(-) 0,7825

(b)

Algoritmos	NDCG@6	NDCG@7	NDCG@8	NDCG@9	NDCG@10
Regression	(-) 0,6393	(-) 0,6536	(-) 0,6536	(-) 0,6536	(-) 0,6536
RankSVM	(-) 0,7957	(-) 0,8005	(-) 0,8005	(-) 0,8047	(-) 0,8062
ListNet	(+) 0,8037	(-) 0,8084	(+) 0,8128	(+) 0,8128	(+) 0,8128
AdaRank-MAP	(-) 0,7413	(-) 0,7431	(-) 0,7497	(-) 0,7497	(-) 0,7497
AdaRank-NDCG	(-) 0,7225	(-) 0,7273	(-) 0,7384	(-) 0,7384	(-) 0,7384
SVMmap	(-) 0,7947	(-) 0,7994	(-) 0,8039	(-) 0,8039	(-) 0,8079
RankSupernova	0,8011	0,8106	0,8122	0,8122	0,8122
RankBoost	(-) 0,6667	(-) 0,681	(-) 0,6854	(-) 0,6854	(-) 0,6914
FRank	(-) 0,6992	(-) 0,7087	(-) 0,7132	(-) 0,7216	(-) 0,7296
Regression+L2reg	(-) 0,7903	(-) 0,7998	(-) 0,7998	(-) 0,804	(-) 0,804
RankSVM-Primal	(-) 0,7816	(-) 0,7864	(-) 0,7908	(-) 0,795	(-) 0,795
RankSVM-Struct	(-) 0,7843	(-) 0,789	(-) 0,7935	(-) 0,7977	(-) 0,7977
SmoothRank	(-) 0,798	(-) 0,8075	(-) 0,8075	(-) 0,8075	(-) 0,8075

Tabla 3.11: Valores del rendimiento alcanzado en la ordenación sobre HP2003, considerando NDCG como medida de desempeño en las posiciones de la 1 a la 10.

(a)

	NDCG@1	NDCG@2	NDCG@3	NDCG@4	NDCG@5
Regression	(-) 0,42	(-) 0,47	(-) 0,5097	(-) 0,5353	(-) 0,5463
RankSVM	(-) 0,6933	(-) 0,7467	(-) 0,7749	(-) 0,7877	(-) 0,7954
ListNet	(+) 0,72	(+) 0,7967	(+) 0,8128	(+) 0,8235	(+) 0,8298

AdaRank-MAP	(+) 0,7333	(+) 0,8	(+) 0,8051	(+) 0,816	(+) 0,8252
AdaRank-NDCG	(=) 0,7133	(+) 0,7667	(+) 0,7902	(+) 0,7946	(+) 0,8006
SVMmap	(=) 0,7133	(-) 0,7567	(+) 0,7791	(-) 0,7898	(-) 0,7922
RankSupernova	0,7133	0,76	0,7785	0,7929	0,7978
RankBoost	(-) 0,6667	(+) 0,77	(+) 0,792	(=) 0,7929	(+) 0,8034
FRank	(-) 0,6533	(-) 0,73	(-) 0,7432	(-) 0,7632	(-) 0,778
Regression+L2reg	(-) 0,6933	(+) 0,7867	(+) 0,8086	(+) 0,8147	(+) 0,8142
RankSVM-Primal	(+) 0,74	(+) 0,7667	(+) 0,7907	(+) 0,8018	(+) 0,8081
RankSVM-Struct	(+) 0,74	(+) 0,77	(+) 0,7907	(+) 0,7996	(+) 0,8074
SmoothRank	(=) 0,7133	(+) 0,78	(+) 0,8083	(+) 0,821	(+) 0,8287

(b)

	NDCG@6	**NDCG@7**	**NDCG@8**	**NDCG@9**	**NDCG@10**
Regression	(-) 0,5511	(-) 0,5689	(-) 0,5711	(-) 0,5795	(-) 0,5943
RankSVM	(+) 0,8015	(-) 0,8048	(-) 0,807	(-) 0,807	(-) 0,8077
ListNet	(+) 0,8317	(+) 0,8349	(+) 0,8349	(+) 0,8357	(+) 0,8372
AdaRank-MAP	(+) 0,8276	(+) 0,8324	(+) 0,8346	(+) 0,8357	(+) 0,8384
AdaRank-NDCG	(+) 0,8038	(-) 0,805	(-) 0,805	(-) 0,805	(-) 0,806
SVMmap	(-) 0,797	(-) 0,7994	(-) 0,7994	(-) 0,7994	(-) 0,7994
RankSupernova	0,8	0,8109	0,8131	0,8163	0,8183
RankBoost	(+) 0,8069	(+) 0,8137	(+) 0,8159	(+) 0,8166	(-) 0,8171
FRank	(-) 0,7879	(-) 0,7891	(-) 0,7891	(-) 0,7909	(-) 0,797
Regression+L2reg	(+) 0,8151	(+) 0,8151	(+) 0,8158	(-) 0,8158	(+) 0,8216
RankSVM-Primal	(+) 0,8129	(+) 0,8162	(+) 0,8162	(+) 0,8172	(-) 0,818
RankSVM-Struct	(+) 0,8096	(+) 0,812	(+) 0,8162	(-) 0,8162	(-) 0,8162
SmoothRank	(+) 0,831	(+) 0,831	(+) 0,831	(+) 0,831	(+) 0,8325

Tabla 3.12: Valores del rendimiento alcanzado en la ordenación sobre HP2004, considerando NDCG como medida de desempeño en las posiciones de la 1 a la 10.

(a)

Algoritmos	**NDCG@1**	**NDCG@2**	**NDCG@3**	**NDCG@4**	**NDCG@5**
Regression	(-) 0,3867	(-) 0,54	(-) 0,5752	(-) 0,6073	(-) 0,613
RankSVM	(-) 0,5733	(-) 0,68	(-) 0,7147	(-) 0,7397	(-) 0,7512
ListNet	(-) 0,6	(-) 0,6867	(-) 0,7213	(-) 0,7618	(-) 0,7694
AdaRank-MAP	(-) 0,6133	(+) 0,7733	(+) 0,8164	(+) 0,8248	(+) 0,8277
AdaRank-NDCG	(-) 0,5867	(-) 0,7333	(-) 0,7512	(+) 0,7862	(-) 0,792
SVMmap	(-) 0,6267	(=) 0,74	(-) 0,7536	(+) 0,7953	(+) 0,8011
RankSupernova	0,64	0,74	0,7578	0,775	0,7951
RankBoost	(-) 0,5067	(-) 0,66	(-) 0,6989	(-) 0,7039	(-) 0,7211
FRank	(-) 0,6	(-) 0,7067	(-) 0,7287	(-) 0,7371	(-) 0,7486

Regression+L2reg	(-) 0,5333	(-) 0,62	(-) 0,6547	(-) 0,6864	(-) 0,6979
RankSVM-Primal	(-) 0,5733	(-) 0,6867	(-) 0,7129	(-) 0,7413	(-) 0,7528
RankSVM-Struct	(-) 0,5867	(-) 0,6733	(-) 0,7248	(-) 0,7465	(-) 0,7523
SmoothRank	(-) 0,6133	(+) 0,7533	(+) 0,7964	(+) 0,8036	(+) 0,8169

(b)

Algoritmos	NDCG@6	NDCG@7	NDCG@8	NDCG@9	NDCG@10
Regression	(-) 0,6285	(-) 0,6285	(-) 0,6351	(-) 0,6428	(-) 0,6468
RankSVM	(-) 0,7589	(-) 0,7589	(-) 0,7634	(-) 0,7647	(-) 0,7687
ListNet	(-) 0,7797	(-) 0,7845	(-) 0,7845	(-) 0,7845	(-) 0,7845
AdaRank-MAP	(+) 0,8328	(+) 0,8328	(+) 0,8328	(+) 0,8328	(+) 0,8328
AdaRank-NDCG	(-) 0,7971	(-) 0,7971	(-) 0,8016	(-) 0,8037	(-) 0,8057
SVMmap	(+) 0,8062	(+) 0,8062	(+) 0,8062	(-) 0,8062	(-) 0,8062
RankSupernova	0,8054	0,8054	0,8054	0,8068	0,8068
RankBoost	(-) 0,7263	(-) 0,7263	(-) 0,7344	(-) 0,7428	(-) 0,7428
FRank	(-) 0,7554	(-) 0,7554	(-) 0,7554	(-) 0,7575	(-) 0,7615
Regression+L2reg	(-) 0,703	(-) 0,7125	(-) 0,7125	(-) 0,7188	(-) 0,7188
RankSVM-Primal	(-) 0,7683	(-) 0,7706	(-) 0,7706	(-) 0,772	(-) 0,772
RankSVM-Struct	(-) 0,7652	(-) 0,7652	(-) 0,7666	(-) 0,7666	(-) 0,7666
SmoothRank	(+) 0,8221	(+) 0,8221	(+) 0,8221	(+) 0,8221	(+) 0,8221

Tabla 3.13: Valores del rendimiento alcanzado en la ordenación sobre TD2003, considerando NDCG como medida de desempeño en las posiciones de la 1 a la 10.

(a)

Algoritmos	NDCG@1	NDCG@2	NDCG@3	NDCG@4	NDCG@5
Regression	(-) 0,32	(-) 0,32	(-) 0,3071	(-) 0,3082	(-) 0,2984
RankSVM	(-) 0,32	(-) 0,33	(-) 0,3441	(+) 0,3531	(+) 0,3621
ListNet	(=) 0,4	(-) 0,34	(-) 0,3365	(-) 0,3254	(-) 0,3393
AdaRank-MAP	(-) 0,26	(-) 0,31	(-) 0,3067	(-) 0,2965	(-) 0,3029
AdaRank-NDCG	(-) 0,36	(-) 0,28	(-) 0,2908	(-) 0,2957	(-) 0,2939
SVMmap	(-) 0,32	(-) 0,34	(-) 0,3199	(+) 0,3377	(-) 0,3318
RankSupernova	0,4	0,35	0,3458	0,3337	0,3426
RankBoost	(-) 0,28	(-) 0,3	(-) 0,3246	(-) 0,3206	(-) 0,3149
FRank	(-) 0,3	(-) 0,28	(-) 0,2671	(-) 0,2565	(-) 0,2468
Regression+L2reg	(-) 0,34	(+) 0,36	(+) 0,3565	(+) 0,3399	(-) 0,3381
RankSVM-Primal	(-) 0,32	(+) 0,37	(+) 0,3554	(+) 0,3631	(+) 0,3659
RankSVM-Struct	(-) 0,34	(-) 0,34	(-) 0,343	(+) 0,3578	(+) 0,3654
SmoothRank	(-) 0,38	(+) 0,36	(-) 0,3319	(+) 0,3346	(-) 0,3345

(b)

Algoritmos	NDCG@6	NDCG@7	NDCG@8	NDCG@9	NDCG@10
Regression	(-) 0,3112	(-) 0,3158	(-) 0,3239	(-) 0,3251	(-) 0,3263
RankSVM	(+) 0,3551	(+) 0,3461	(+) 0,344	(+) 0,3451	(+) 0,3461
ListNet	(+) 0,3478	(+) 0,3542	(+) 0,3481	(+) 0,3501	(+) 0,3484
AdaRank-MAP	(-) 0,308	(-) 0,3106	(-) 0,3089	(-) 0,3062	(-) 0,3069
AdaRank-NDCG	(-) 0,2948	(-) 0,3031	(-) 0,301	(-) 0,303	(-) 0,3036
SVMmap	(-) 0,3272	(-) 0,3253	(-) 0,3232	(-) 0,3229	(-) 0,3282
RankSupernova	0,3432	0,3352	0,3275	0,3304	0,3371
RankBoost	(-) 0,3204	(-) 0,3134	(-) 0,3132	(-) 0,3116	(-) 0,3122
FRank	(-) 0,2537	(-) 0,2538	(-) 0,2548	(-) 0,2654	(-) 0,269
Regression+L2reg	(-) 0,3264	(-) 0,3324	(+) 0,3319	(-) 0,3277	(-) 0,3297
RankSVM-Primal	(+) 0,3625	(+) 0,3579	(+) 0,3523	(+) 0,3548	(+) 0,3571
RankSVM-Struct	(+) 0,3584	(+) 0,356	(+) 0,3478	(+) 0,3497	(+) 0,3467
SmoothRank	(-) 0,3328	(+) 0,3363	(+) 0,3358	(+) 0,3339	(-) 0,3367

Tabla 3.14: Valores del rendimiento alcanzado en la ordenación sobre TD2004, considerando NDCG como medida de desempeño en las posiciones de la 1 a la 10.

(a)

Algoritmos	NDCG@1	NDCG@2	NDCG@3	NDCG@4	NDCG@5
Regression	(-) 0,36	(-) 0,34	(-) 0,3352	(-) 0,3281	(-) 0,3257
RankSVM	(-) 0,4133	(-) 0,3467	(-) 0,3467	(-) 0,341	(-) 0,324
ListNet	(-) 0,36	(-) 0,3467	(-) 0,3573	(-) 0,3469	(-) 0,3325
AdaRank-MAP	(-) 0,4133	(=) 0,3933	(+) 0,3757	(+) 0,3683	(+) 0,3602
AdaRank-NDCG	(=) 0,4267	(-) 0,38	(-) 0,3688	(-) 0,3524	(-) 0,3514
SVMmap	(-) 0,2933	(-) 0,3067	(-) 0,3035	(-) 0,2997	(-) 0,3007
RankSupernova	0,4267	0,3933	0,3725	0,3646	0,3578
RankBoost	(+) 0,5067	(+) 0,4333	(+) 0,4295	(+) 0,4053	(+) 0,3878
FRank	(+) 0,4933	(+) 0,4067	(+) 0,3875	(-) 0,3581	(+) 0,3629
Regression+L2reg	(-) 0,2933	(-) 0,32	(-) 0,304	(-) 0,2964	(-) 0,2917
RankSVM-Primal	(-) 0,3067	(-) 0,3067	(-) 0,3131	(-) 0,3026	(-) 0,3062
RankSVM-Struct	(-) 0,3467	(-) 0,3467	(-) 0,3371	(-) 0,3287	(-) 0,3192
SmoothRank	(-) 0,4	(+) 0,42	(+) 0,3832	(+) 0,3696	(-) 0,3555

(b)

Algoritmos	NDCG@6	NDCG@7	NDCG@8	NDCG@9	NDCG@10
Regression	(-) 0,3134	(-) 0,3075	(-) 0,3087	(-) 0,3061	(-) 0,3031
RankSVM	(-) 0,3183	(-) 0,3158	(-) 0,3105	(-) 0,3063	(-) 0,3078
ListNet	(-) 0,327	(-) 0,3251	(-) 0,3206	(-) 0,3188	(-) 0,3175
AdaRank-MAP	(-) 0,3533	(+) 0,3438	(-) 0,3362	(-) 0,3341	(-) 0,3285

AdaRank-NDCG	(-) 0,3376	(-) 0,3297	(-) 0,3284	(-) 0,3212	(-) 0,3163
SVMmap	(-) 0,2942	(-) 0,2979	(-) 0,2934	(-) 0,2906	(-) 0,2907
RankSupernova	0,3549	0,3421	0,3381	0,3366	0,3302
RankBoost	(+) 0,3765	(+) 0,3639	(+) 0,3594	(+) 0,3527	(+) 0,3504
FRank	(=) 0,3549	(+) 0,3474	(+) 0,3392	(-) 0,3334	(+) 0,3331
Regression+L2reg	(-) 0,2897	(-) 0,2902	(-) 0,2858	(-) 0,2844	(-) 0,2832
RankSVM-Primal	(-) 0,3003	(-) 0,2951	(-) 0,2921	(-) 0,2942	(-) 0,2913
RankSVM-Struct	(-) 0,3129	(-) 0,3102	(-) 0,3146	(-) 0,3084	(-) 0,309
SmoothRank	(-) 0,3426	(+) 0,344	(+) 0,3432	(+) 0,3394	(+) 0,3343

3.4.2 Análisis estadístico

Después de haber obtenido los resultados del desempeño del método RankSupernova con respecto a los métodos de referencia, se realizaron pruebas estadísticas para fundamentar con un basamento matemático el nivel de significación y comportamiento reflejados por la precisión alcanzada en la ordenación a nivel de consulta.

Para cada conjunto de datos se obtuvo la predicción realizada por el método a cada consulta y posteriormente, con el empleo de una de las herramientas de evaluación propuestas por LETOR: Eval-Score-3.0.pl [46], se generaron los ficheros que muestran el desempeño en cuanto a precisión alcanzada por el algoritmo a nivel de consulta, en relación a las medidas de evaluación MAP, P@n y NDCG.

Se tomaron los valores de precisión en la ordenación determinados por cada algoritmo para MAP y se aplicaron pruebas paramétricas utilizando un modelo estadístico de comparación de medias para muestras relacionadas o pareadas. Este modelo fue aplicado porque se ajusta bien a la necesidad de comparar diferentes algoritmos en un mismo grupo de consultas y permite llevar a cabo un estudio comparativo de los desempeños de cada algoritmo con un alto nivel de veracidad y exactitud.

Con el empleo de SPSS[6], se realizó un análisis de varianza (ANOVA) de un factor con medidas repetidas (MR); el factor que se toma es *algoritmo* (Ver Tabla 3.15), con 8 niveles y como variable dependiente el rendimiento alcanzado en la

[6] Statistical Product and Service Solutions http://www.spss.com/

ordenación sobre cada conjunto de datos estudiado: OHSUMED, NP2003, NP2004, HP2003, HP2004, TD2003 y TD2004, considerando a MAP como medida de desempeño.

Tabla 3.15: Niveles del factor algoritmo con su numeración correspondiente

Numeración	Algoritmo
1	AdaRankMAP
2	AdaRankNDCG
3	FRank
4	ListNet
5	RankBoost
6	RankSupernova
7	RankSVM
8	Regression

Inicialmente, se determinaron para cada uno de los conjuntos de datos sus principales descriptores asociados: media estimada, error estándar y el intervalo de confianza. Las Tablas 3.16, 3.18, 3.20, 3.22, 3.24, 3.26 y 3.28, muestran estos valores sobre la base del desempeño de los métodos analizados.

Con el objetivo de contrastar el efecto multivariado del factor *algoritmo* en cada uno de los conjuntos de datos. Se aplicaron las pruebas: Traza de Pillai, Lambda de Wilks, Traza de Hotelling y Raíz mayor de Roy (Ver las Tablas 3.17, 3.19, 3.21, 3.23, 3.25, 3.27 y 3.29). Estos contrastes se basan en las comparaciones por pares, linealmente independientes, entre las medias marginales estimadas; utilizando un estadístico exacto y calculado con *alfa* = 0.05. Para una descripción de estos estadísticos puede consultarse Bock [47] o Tabachnik y Fidel [48].

Para la interpretación de los resultados se toma el valor del nivel crítico (Sig.), si es menor que 0.05 (valor de *alfa*) se rechaza la hipótesis nula de igualdad de medias y se concluye que el rendimiento alcanzado en la ordenación, considerando a MAP como medida de desempeño, no es el mismo en los ocho algoritmos de Aprendizaje de la Ordenación definidos por el factor *algoritmo*.

Para todos los conjuntos de datos analizados (excepto para TD2003), se comprobó que existen diferencias significativas entre los algoritmos de Aprendizaje de la Ordenación confrontados.

Para conocer tales diferencias, se aplicaron comparaciones a pares basadas en las medidas marginales estimadas, empleando el ajuste para comparaciones múltiples: Bonferroni; considerando los principales descriptores asociados y la diferencia de medias significativa al nivel 0.05.

Los resultados de estas comparaciones a pares, considerando el desempeño de los métodos a nivel de consulta pueden ser analizados en el Anexo A.

Finalmente, se obtuvo que RankSupernova supera significativamente al método Regression en los conjuntos de datos: OHSUMED, NP2003, NP2004, HP2003 y HP2004. Alcanza valores significativos comparado con RankSVM en OHSUMED y con AdaRankNDCG en TD2004. Además, se puede constatar que el método propuesto no es superado significativamente por ningún método de aprendizaje y su valor de media siempre se encuentra entre los tres mejores.

A continuación se muestran los resultados estadísticos para cada uno de los siete conjuntos de datos estudiados.

Tabla 3.16: Estimaciones obtenidas a partir del rendimiento de los métodos sobre OHSUMED

algoritmo	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
1	,448	,022	,405	,491
2	,449	,022	,406	,492
3	,443	,021	,401	,485
4	,445	,022	,402	,488
5	,440	,022	,396	,484
6	,451	,021	,409	,494
7	,432	,022	,389	,475
8	,421	,021	,380	,462

Tabla 3.17: Resultado de los Test multivariados aplicados al desempeño de los métodos de aprendizaje sobre OHSUMED

	Value	F	Hypothesis df	Error df	Sig.	Noncent. Parameter	Observed Power[b]
Pillai's trace	,210	3,754[a]	7,000	99,000	,001	26,275	,971
Wilks' lambda	,790	3,754[a]	7,000	99,000	,001	26,275	,971
Hotelling's trace	,265	3,754[a]	7,000	99,000	,001	26,275	,971
Roy's largest root	,265	3,754[a]	7,000	99,000	,001	26,275	,971

a. Exact statistic

b. Computed using alpha = ,05

Tabla 3.18: Estimaciones obtenidas a partir del rendimiento de los métodos sobre NP2003

algoritmo	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
1	,678	,032	,615	,742
2	,668	,032	,605	,731
3	,664	,031	,602	,726
4	,690	,030	,629	,750
5	,707	,031	,646	,768
6	,691	,031	,629	,752
7	,696	,031	,635	,756
8	,564	,033	,498	,630

Tabla 3.19: Resultado de los Test multivariados aplicados al desempeño de los métodos de aprendizaje sobre NP2003

	Value	F	Hypothesis df	Error df	Sig.	Noncent. Parameter	Observed Power[b]
Pillai's trace	,197	5,008[a]	7,000	143,000	,000	35,059	,996
Wilks' lambda	,803	5,008[a]	7,000	143,000	,000	35,059	,996
Hotelling's trace	,245	5,008[a]	7,000	143,000	,000	35,059	,996
Roy's largest root	,245	5,008[a]	7,000	143,000	,000	35,059	,996

a. Exact statistic

b. Computed using alpha = ,05

Tabla 3.20: Estimaciones obtenidas a partir del rendimiento de los métodos sobre NP2004

algoritmo	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
1	,622	,045	,532	,712
2	,627	,046	,535	,719
3	,601	,046	,509	,692
4	,652	,044	,564	,740
5	,564	,047	,471	,657
6	,675	,043	,589	,761
7	,659	,043	,573	,745
8	,514	,045	,424	,605

Tabla 3.21: Resultado de los Test multivariados aplicados al desempeño de los métodos de aprendizaje sobre NP2004

	Value	F	Hypothesis df	Error df	Sig.	Noncent. Parameter	Observed Power[b]
Pillai's trace	,272	3,638[a]	7,000	68,000	,002	25,466	,961
Wilks' lambda	,728	3,638[a]	7,000	68,000	,002	25,466	,961
Hotelling's trace	,375	3,638[a]	7,000	68,000	,002	25,466	,961
Roy's largest root	,375	3,638[a]	7,000	68,000	,002	25,466	,961

a. Exact statistic

b. Computed using alpha = ,05

Tabla 3.22: Estimaciones obtenidas a partir del rendimiento de los métodos sobre HP2003

algoritmo	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
1	,771	,028	,715	,827
2	,748	,030	,689	,807
3	,709	,030	,650	,769
4	,766	,028	,711	,821
5	,733	,029	,675	,791
6	,753	,029	,694	,811
7	,741	,029	,683	,799
8	,497	,033	,432	,562

Tabla 3.23: Resultado de los Test multivariados aplicados al desempeño de los métodos de aprendizaje sobre HP2003

	Value	F	Hypothesis df	Error df	Sig.	Noncent. Parameter	Observed Power[b]
Pillai's trace	,378	12,395[a]	7,000	143,000	,000	86,764	1,000
Wilks' lambda	,622	12,395[a]	7,000	143,000	,000	86,764	1,000
Hotelling's trace	,607	12,395[a]	7,000	143,000	,000	86,764	1,000
Roy's largest root	,607	12,395[a]	7,000	143,000	,000	86,764	1,000

Each F tests the multivariate effect of algoritmo. These tests are based on the linearly independent pairwise comparisons among the estimated marginal means.

a. Exact statistic

b. Computed using alpha = ,05

Tabla 3.24: Estimaciones obtenidas a partir del rendimiento de los métodos sobre HP2004

algoritmo	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
1	,740	,041	,657	,822
2	,693	,042	,608	,777
3	,686	,046	,594	,778
4	,668	,046	,577	,759
5	,628	,045	,539	,717
6	,715	,046	,624	,805
7	,642	,046	,551	,733
8	,534	,046	,442	,625

Tabla 3.25: Resultado de los Test multivariados aplicados al desempeño de los métodos de aprendizaje sobre HP2004

	Value	F	Hypothesis df	Error df	Sig.	Noncent. Parameter	Observed Power[b]
Pillai's trace	,322	4,606[a]	7,000	68,000	,000	32,242	,990
Wilks' lambda	,678	4,606[a]	7,000	68,000	,000	32,242	,990
Hotelling's trace	,474	4,606[a]	7,000	68,000	,000	32,242	,990
Roy's largest root	,474	4,606[a]	7,000	68,000	,000	32,242	,990

a. Exact statistic

b. Computed using alpha = ,05

Tabla 3.26: Estimaciones obtenidas a partir del rendimiento de los métodos sobre TD2003

algoritmo	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
1	,228	,032	,165	,292
2	,237	,033	,170	,303
3	,203	,029	,146	,260
4	,275	,032	,211	,340
5	,227	,023	,182	,273
6	,259	,032	,195	,322
7	,263	,029	,204	,321
8	,241	,029	,182	,300

Tabla 3.27: Resultado de los Test multivariados aplicados al desempeño de los métodos de aprendizaje sobre TD2003

	Value	F	Hypothesis df	Error df	Sig.	Noncent. Parameter	Observed Power[b]
Pillai's trace	,211	1,640[a]	7,000	43,000	,150	11,482	,604
Wilks' lambda	,789	1,640[a]	7,000	43,000	,150	11,482	,604
Hotelling's trace	,267	1,640[a]	7,000	43,000	,150	11,482	,604
Roy's largest root	,267	1,640[a]	7,000	43,000	,150	11,482	,604

a. Exact statistic

b. Computed using alpha = ,05

Tabla 3.28: Estimaciones obtenidas a partir del rendimiento de los métodos sobre TD2004

algoritmo	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
1	,219	,020	,180	,258
2	,194	,016	,161	,226
3	,239	,019	,202	,276
4	,223	,019	,186	,261
5	,261	,020	,221	,301
6	,231	,020	,191	,270
7	,224	,018	,188	,259
8	,208	,016	,177	,239

Tabla 3.29: Resultado de los Test multivariados aplicados al desempeño de los métodos de aprendizaje sobre TD2004

	Value	F	Hypothesis df	Error df	Sig.	Noncent. Parameter	Observed Power[b]
Pillai's trace	,322	4,607[a]	7,000	68,000	,000	32,251	,990
Wilks' lambda	,678	4,607[a]	7,000	68,000	,000	32,251	,990
Hotelling's trace	,474	4,607[a]	7,000	68,000	,000	32,251	,990
Roy's largest root	,474	4,607[a]	7,000	68,000	,000	32,251	,990

a. Exact statistic

b. Computed using alpha = ,05

3.5 Conclusiones

Después de haber realizado un estudio experimental para evaluar el desempeño del método propuesto, considerando para ello el rendimiento alcanzado en la ordenación para las colecciones de datos de Letor 3.0; se pudo demostrar que

RankSupernova manifiesta un alto nivel competitivo con respecto a los principales métodos referenciados en la literatura.

En este sentido, se realizó un análisis estadístico, donde los resultados obtenidos permiten afirmar que RankSupernova presenta mejoras significativas en cuanto a precisión en la ordenación en comparación con los métodos Regression, RankSVM y AdaRank-NDCG.

Finalmente se puede concluir que para las colecciones de datos OHSUMED y Gov, los métodos que optimizan directamente medidas de desempeño mejoran de forma significativa el desempeño de los métodos tradicionales; sin embargo, en la mayoría de los casos, estos no presentan diferencias significativas entre sí.

Conclusiones

En la presente investigación se evidencia como los métodos tradicionales de Recuperación de Información, han ido cediendo su espacio a nuevas técnicas de aprendizaje automático, debido a que los nuevos problemas de ordenación requieren soluciones y modelos de gran precisión que permitan mejorar la satisfacción de los usuarios.

Con este propósito surgió el Aprendizaje de la Ordenación que ha venido a desempeñar un rol vital para muchas aplicaciones. En este trabajo, enfocando el problema de la ordenación en la recuperación de documentos, se han presentado y descrito las diferentes categorías en las que se enmarcan los métodos desarrollados para dar solución a esta problemática.

Se presentó un nuevo método de Aprendizaje de la Ordenación para la Recuperación de Información, utilizando la optimización directa de medidas de desempeño y basándose en el algoritmo de optimización global Supernova. Este método propuesto, denominado RankSupernova, puede optimizar directamente cualquier medida de desempeño utilizada en la R.I.

En el estudio experimental para evaluar el desempeño de RankSupernova, considerando para ello el rendimiento alcanzado en la ordenación para las colecciones de datos OHSUMED y Gov, se pudo demostrar como el método propuesto manifiesta un alto nivel competitivo en relación a los principales métodos referenciados en la literatura.

Se realizó además, todo un análisis estadístico, donde los resultados obtenidos permiten afirmar que RankSupernova presenta mejoras significativas en cuanto a precisión en la ordenación en comparación con los métodos Regression, RankSVM y AdaRank-NDCG; además, no es superado significativamente por ningún otro método de Aprendizaje de la Ordenación y el valor de su media se ubica entre las tres mejores posiciones para todos los casos.

Considerando esto, se puede afirmar finalmente que para las colecciones de datos de LETOR 3.0 los métodos que optimizan directamente medidas de desempeño mejoran de forma significativa el desempeño de las propuestas tradicionales; sin embargo, estos en la mayoría de los casos no presentan diferencias significativas entre sí.

Recomendaciones

Dentro de los principales trabajos que pueden ser dirigidos a partir de la presente investigación están:

- ✓ Conducir más experimentos con nuevas y referenciadas colecciones de datos de mediano y gran tamaño para validar la eficiencia, y evaluar el desempeño del método RankSupernova bajo disímiles condiciones.
- ✓ Utilizar otras medidas de desempeño, por ejemplo, P@n y NDCG; como función general *E*, con el objetivo de evaluar el comportamiento de la precisión en la ordenación, considerando el mismo diseño experimental conformado para MAP.
- ✓ Aplicar otras variantes de error en la función de pérdida básica, como por ejemplo, el Error Cuadrático Medio.
- ✓ Rediseñar la función de *ranking* considerando las relaciones y el comportamiento de los rasgos.
- ✓ Emplear algoritmos adaptativos para tratar el problema de ajustar los parámetros de RankSupernova. Debido a que ajustar los parámetros de forma empírica (manualmente) es usualmente difícil, especialmente cuando hay muchos parámetros y las medidas de evaluación no son uniformes.
- ✓ Desarrollar estrategias multi-supernovas basadas en la concepción de RankSupernova, con la intención de mejorar el rendimiento y disminuir el tiempo computacional.

Bibliografía

[1] Alejo, Oscar J., Fernández, Juan M., Huete, Juan F., y Pérez, Ramiro, «Optimización directa de Medidas de Desempeño en el Aprendizaje de la Ordenación usando Enjambre de Partículas», Diploma de Estudios Avanzados del Doctorado en Soft Computing, Universidad de Granada, 2010.

[2] Mesa, Eddy, Velásquez, Juan D., y Jaramillo, Gloria, «Supernova: un algoritmo novedoso de optimización global», Tesis de Maestría, Universidad Nacional de Colombia, 2010.

Referencias bibliográficas

[1] Salton, G. y McGill, M. H., *Introduction to modern information retrieval*. New York: McGraw-Hill, 1984.

[2] Salvador Oliván, J.A. y Arquero Áviles, R., «Una aproximación al concepto de Recuperación de Información en el marco de la ciencia de la documentación», *Investigación Bibliotecológica*, vol. 20, págs. 13-43, Dic. 2006.

[3] Baeza-Yates, R. y Ribeiro-Neto, B., *Modern information retrieval*. NewYork: Addison-Wesley, 1999.

[4] Salvador Oliván, J.A. y Arquero Avilés, R., «La investigación en Recuperación de Información: Revisión de tendencias actuales y críticas», *Cuadernos de Documentación Multimedia*, vol. 15, 2004.

[5] Salton, G., E.A.F., y Wu, H., «Extended Boolean information retrieval», *Proceedings of the Communications of the ACM*, vol. 26, nº. 11, págs. 1022-1036, 1983.

[6] Wong, S.K., W.Z., y Wong, P.C., «Generalized vector space model in information retrieval», *Proceedings of the Eighth Annual ACM-SIGIR Conference on Research and Development in Information Retrieval, Association for Computing Machinery*, págs. 18-25, 1985.

[7] Croft, W.B. y D.J.H., «Using probabilistic models of information retrieval without relevance information», *Proceedings of the Journal of Documentation*, vol. 35, nº. 4, págs. 285-295, 1975.

[8] Robertson, S., S.W., Jones, S., Hancock-Beaulieu, M., y Gatford, M., «Okapi at TREC-3», *Proceedings of TREC-3*, 1995.

[9] Manning, C.D. y H.S., «Foundations of Statistical Language Processing», *The MIT Press*, 1999.

[10] Deerwester, S.C., Landauer, T.K., Furnas. G.W., y Harshman R.A., «Indexing by latent semantic analysis», *Proceedings of the Journal of the American Society of Information Science*, vol. 41, nº. 6, págs. 391-407, 1990.

[11] Lewis, D.D. y Spärck Jones, K., «Natural language processing for information retrieval», *Proceedings of CACM*, vol. 39, nº. 1, págs. 92-101, 1996.

[12] Brin, S. y L.P., «The anatomy of a large-scale hypertextual web search engine», *Proceedings of WWW*, págs. 107-117, 1998.

[13] Gyongyi, Z., H.G.M., y Pedersen, J., «Combating web spam with TrustRank», *Proceedings of the 30th VLDB Conference*, 2004.

[14] Liu, Y. et al., «BrowseRank: letting web users vote for page importance», *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 2008.

[15] «MODELOS DE RECUPERACION DE INFORMACION», 14-Nov-2011. [Online]. Available: http://modelosrecuperacion.freeservers.com/. [Accessed: 14-Nov-2011].

[16] Alejo, Oscar J., Fernández, Juan M., Huete, Juan F., y Pérez, Ramiro, «Optimización directa de Medidas de Desempeño en el Aprendizaje de la Ordenación usando Enjambre de Partículas», Diploma de Estudios Avanzados del Doctorado en Soft Computing, Universidad de Granada, 2010.

[17] Tie-Yang, Liu, J.X, Qin, T., W.-Y., Xiong, y Li, H., «Letor: Benchmark dataset for research on learning to rank for information retrieval», *Proceedings of SIGIR*, 2007.

[18] K. Jarvelin, A., «Cumulated Gain-Based Evaluation of IR Techniques», *Proceedings of ACM Transactions on Information Systems*, 2002.

[19] Vargas Sandoval, S. (último), «Learning to Rank: Técnicas de Aprendizaje Automático en Recuperación de Información.» 21-Jun-2011.

[20] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, y Hang Li, «Learning to Rank: From Pairwise Approach to Listwise Approach», *Proceedings of the 24th International Conference on Machine Learning*, 2007.

[21] Herbrich, R., Graepel, T., y Obermayer, K., «Large margin rank boundaries for ordinal regression», *Proceedings of the Advances in Large Margin Classifiers*, 2000.

[22] Freund, Y., R.I., Schapire, R., y Singer, Y., «An efficient boosting algorithm for combining preferences», *Proceedings of JMLR*, págs. 933-969, 2003.

[23] Hamed Valizadegan, Rong Jin, Ruofei Zhang, y Jianchang Mao, «Learning to Rank by Optimizing NDCG Measure». .

[24] Joachims, T., «A support vector method for multivariate performance measures», *Proceedings of ICML*, 2005.

[25] Yue, Y., Radlinski, F., F. T., y Joachims, T., «A support vector method for optimizing average precision», *Proceedings of SIGIR*, 2007.

[26] Chapelle, O., Smola, A., y L. Q., «Large Margin Optimization of Ranking Measures», *Proceedings of NIPS*, 2007.

[27] Xu, J., Liu, T-Y, Lu, M., Li, H., y Ma, W-Y., «Directly Optimizing Evaluation Measures in Learning to Rank», *Proceedings of SIGIR*, 2008.

[28] Zheng, Z., H.Z., Zhang, T., Chapelle, O., Chen, K., y Sun, G., «A General Boosting Method and its Application to Learning Ranking Functions for Web Search», *Proceedings of NIPS*, 2007.

[29] Xu, J. y Li, H., «Adarank: a boosting algorithm for information retrieval», *Proceedings of SIGIR*, 2007.

[30] Burges, C., R.R., y Le, Q., «Learning to rank with nonsmooth cost functions», *Proceedings of NIPS*, 2006.

[31] Taylor, M., J.G., Robertson, S., y Minka, T., «Softrank: Optimizing non-smooth rank metrics», *Proceedings of SIGIR*, 2007.

[32] Fan, W., P.P., y Wallace, L., «Nonlinear ranking function representations in genetic programming-based ranking discovery for personalized search», *Proceedings of Decision Support Systems*, 2006.

[33] Yeh, J-Y, Liu, J-Y, Ke, H-R, y Chang, W-P, «Learning to rank for information retrieval using genetic programming», *Proceedings of SIGIR*, 2007.

[34] Cossock, D. y T.Z., «Subset ranking using regression», *Proceedings of COLT*, 2006.

[35] Metzler, D., W.C., y McCallum, A., «Direct maximization of rank-based metrics for information retrieval», *Proceedings of Technical report, CIIR*, 2005.

[36] Chapelle, O., «Direct Optimization for Web Search Ranking», *Proceedings of SIGIR*, 2009.

[37] Mesa, Eddy, Velásquez, Juan D., y Jaramillo, Gloria, «Supernova: un algoritmo novedoso de optimización global», Tesis de Maestría, Universidad Nacional de Colombia, 2010.

[38] Carroll, B. y Ostlie, D., *An Introduction to Modern Astrophysic*. Pearson, 2007.

[39] Alonso, M. y Finn, E., *Física*. Addison-Wesley, 1995.

[40] Lieddle, A., *Introduction to Modern Cosmology*. Wiley, 2003.

[41] Zayas Agüero, Pedro M., *El rombo de las investigaciones de las Ciencias Sociales*. .

[42] Bijarro Hernández, Francisco, *Desarrollo estratégico para la Investigación Científica*. eumed.net.

[43] Moreno, Eleazar y Alejo, Oscar J., «L2RLab: Herramienta Informática para asistir el proceso de experimentación y análisis de modelos de Aprendizaje de la Ordenación.», Trabajo de diploma, Universidad de Cienfuegos, 2012.

[44] Sione, Palu, «Java in Soft Computing», 29-Abr-2002. [Online]. Available: http://www.developer.com/Java-in-Soft-Computing.html. [Accessed: 15-Mar-2012].

[45] Zhang, Min, Kuang, Da, Hua, G., Liu, Y., y Ma, S., «Is learning to rank effective for Web search?», *Proceedings of SIGIR*, 2009.

[46] Xu, J., Tie-Yang, Liu, y Hang Li, «The Evaluation Tool in LETOR». [Online]. Available: https://research.microsoft.com/enus/um/beijing/projects/letor/LETOR3.0/EvaluationTool.zip.

[47] Bock, R.D., *Multivariate statistical methods in behavioral research*. New York: McGraw-Hill, 1975.

[48] Tabachnik, B.G. y L.B.F., *Using multivariate statistics*. New York: Harper and Row, 1983.

Anexos

A: Comparaciones a pares considerando el desempeño de los métodos.

A.1: Comparaciones a pares, considerando el desempeño de los métodos sobre OHSUMED.

(I) algoritmo	(J) algoritmo	Mean Difference (I-J)	Std. Error	Sig.[a]	95% Confidence Interval for Difference[a]	
					Lower Bound	Upper Bound
1	2	-,001	,003	1,000	-,010	,008
	3	,005	,004	1,000	-,009	,019
	4	,003	,003	1,000	-,005	,011
	5	,008	,004	1,000	-,005	,020
	6	-,003	,003	1,000	-,012	,005
	7	,016	,005	,062	,000	,031
	8	,027*	,007	,008	,004	,050
2	1	,001	,003	1,000	-,008	,010
	3	,006	,004	1,000	-,006	,018
	4	,004	,002	1,000	-,003	,011
	5	,009	,004	,608	-,003	,021
	6	-,002	,002	1,000	-,010	,005
	7	,017*	,005	,023	,001	,032
	8	,028*	,007	,004	,005	,051
3	1	-,005	,004	1,000	-,019	,009
	2	-,006	,004	1,000	-,018	,006
	4	-,002	,004	1,000	-,015	,011
	5	,003	,005	1,000	-,012	,018
	6	-,008	,004	1,000	-,021	,004
	7	,011	,006	1,000	-,009	,030
	8	,022	,008	,187	-,003	,047
4	1	-,003	,003	1,000	-,011	,005
	2	-,004	,002	1,000	-,011	,003
	3	,002	,004	1,000	-,011	,015
	5	,005	,004	1,000	-,009	,018
	6	-,006	,003	,902	-,016	,003
	7	,013	,005	,292	-,003	,028
	8	,024*	,007	,025	,001	,046
5	1	-,008	,004	1,000	-,020	,005
	2	-,009	,004	,608	-,021	,003
	3	-,003	,005	1,000	-,018	,012
	4	-,005	,004	1,000	-,018	,009
	6	-,011	,004	,180	-,024	,002
	7	,008	,006	1,000	-,011	,027
	8	,019	,008	,338	-,005	,043
6	1	,003	,003	1,000	-,005	,012
	2	,002	,002	1,000	-,005	,010
	3	,008	,004	1,000	-,004	,021
	4	,006	,003	,902	-,003	,016
	5	,011	,004	,180	-,002	,024
	7	,019*	,005	,020	,002	,037
	8	,030*	,007	,001	,007	,053
7	1	-,016	,005	,062	-,031	,000
	2	-,017*	,005	,023	-,032	-,001
	3	-,011	,006	1,000	-,030	,009
	4	-,013	,005	,292	-,028	,003
	5	-,008	,006	1,000	-,027	,011
	6	-,019*	,005	,020	-,037	-,002
	8	,011	,006	1,000	-,009	,032
8	1	-,027*	,007	,008	-,050	-,004
	2	-,028*	,007	,004	-,051	-,005
	3	-,022	,008	,187	-,047	,003
	4	-,024*	,007	,025	-,046	-,001
	5	-,019	,008	,338	-,043	,005
	6	-,030*	,007	,001	-,053	-,007
	7	-,011	,006	1,000	-,032	,009

A.2: Comparaciones a pares, considerando el desempeño de los métodos sobre NP2003

(I) algoritmo	(J) algoritmo	Mean Difference (I-J)	Std. Error	Sig.[a]	95% Confidence Interval for Difference[a]	
					Lower Bound	Upper Bound
1	2	,010	,013	1,000	-,031	,052
	3	,014	,024	1,000	-,061	,090
	4	-,011	,019	1,000	-,071	,049
	5	-,029	,021	1,000	-,097	,039
	6	-,012	,021	1,000	-,078	,053
	7	-,017	,019	1,000	-,079	,044
	8	,114*	,024	,000	,038	,190
2	1	-,010	,013	1,000	-,052	,031
	3	,004	,023	1,000	-,069	,077
	4	-,022	,019	1,000	-,082	,039
	5	-,040	,021	1,000	-,106	,027
	6	-,023	,020	1,000	-,087	,042
	7	-,028	,018	1,000	-,086	,031
	8	,103*	,023	,000	,031	,175
3	1	-,014	,024	1,000	-,090	,061
	2	-,004	,023	1,000	-,077	,069
	4	-,026	,020	1,000	-,088	,037
	5	-,043	,020	,895	-,107	,020
	6	-,027	,024	1,000	-,103	,050
	7	-,032	,020	1,000	-,096	,032
	8	,100*	,028	,016	,010	,189
4	1	,011	,019	1,000	-,049	,071
	2	,022	,019	1,000	-,039	,082
	3	,026	,020	1,000	-,037	,088
	5	-,018	,019	1,000	-,080	,044
	6	-,001	,018	1,000	-,057	,055
	7	-,006	,015	1,000	-,055	,043
	8	,125*	,026	,000	,043	,207
5	1	,029	,021	1,000	-,039	,097
	2	,040	,021	1,000	-,027	,106
	3	,043	,020	,895	-,020	,107
	4	,018	,019	1,000	-,044	,080
	6	,017	,024	1,000	-,058	,092
	7	,012	,019	1,000	-,048	,072
	8	,143*	,027	,000	,058	,228
6	1	,012	,021	1,000	-,053	,078
	2	,023	,020	1,000	-,042	,087
	3	,027	,024	1,000	-,050	,103
	4	,001	,018	1,000	-,055	,057
	5	-,017	,024	1,000	-,092	,058
	7	-,005	,021	1,000	-,071	,061
	8	,126*	,028	,000	,038	,215
7	1	,017	,019	1,000	-,044	,079
	2	,028	,018	1,000	-,031	,086
	3	,032	,020	1,000	-,032	,096
	4	,006	,015	1,000	-,043	,055
	5	-,012	,019	1,000	-,072	,048
	6	,005	,021	1,000	-,061	,071
	8	,131*	,025	,000	,052	,210
8	1	-,114*	,024	,000	-,190	-,038
	2	-,103*	,023	,000	-,175	-,031
	3	-,100*	,028	,016	-,189	-,010
	4	-,125*	,026	,000	-,207	-,043
	5	-,143*	,027	,000	-,228	-,058
	6	-,126*	,028	,000	-,215	-,038
	7	-,131*	,025	,000	-,210	-,052

Based on estimated marginal means

a. Adjustment for multiple comparisons: Bonferroni.

*. The mean difference is significant at the ,05 level.

A.3: Comparaciones a pares, considerando el desempeño de los métodos sobre NP2004

(I) algoritmo	(J) algoritmo	Mean Difference (I-J)	Std. Error	Sig.[a]	95% Confidence Interval for Difference[a]	
					Lower Bound	Upper Bound
1	2	-,005	,025	1,000	-,085	,075
	3	,021	,043	1,000	-,117	,160
	4	-,030	,031	1,000	-,131	,071
	5	,058	,040	1,000	-,072	,188
	6	-,053	,034	1,000	-,162	,056
	7	-,037	,034	1,000	-,147	,074
	8	,108	,038	,160	-,015	,231
2	1	,005	,025	1,000	-,075	,085
	3	,026	,043	1,000	-,114	,166
	4	-,025	,031	1,000	-,124	,074
	5	,063	,039	1,000	-,065	,191
	6	-,048	,034	1,000	-,159	,063
	7	-,032	,033	1,000	-,139	,075
	8	,113	,040	,175	-,017	,242
3	1	-,021	,043	1,000	-,160	,117
	2	-,026	,043	1,000	-,166	,114
	4	-,051	,041	1,000	-,184	,082
	5	,037	,043	1,000	-,104	,177
	6	-,074	,041	1,000	-,208	,059
	7	-,058	,041	1,000	-,190	,074
	8	,087	,049	1,000	-,072	,245
4	1	,030	,031	1,000	-,071	,131
	2	,025	,031	1,000	-,074	,124
	3	,051	,041	1,000	-,082	,184
	5	,088	,037	,521	-,031	,206
	6	-,023	,026	1,000	-,109	,062
	7	-,007	,026	1,000	-,091	,077
	8	,138*	,041	,034	,005	,270
5	1	-,058	,040	1,000	-,188	,072
	2	-,063	,039	1,000	-,191	,065
	3	-,037	,043	1,000	-,177	,104
	4	-,088	,037	,521	-,206	,031
	6	-,111	,037	,101	-,231	,009
	7	-,095	,043	,863	-,234	,045
	8	,050	,046	1,000	-,100	,200
6	1	,053	,034	1,000	-,056	,162
	2	,048	,034	1,000	-,063	,159
	3	,074	,041	1,000	-,059	,208
	4	,023	,026	1,000	-,062	,109
	5	,111	,037	,101	-,009	,231
	7	,016	,025	1,000	-,066	,098
	8	,161*	,035	,000	,049	,274
7	1	,037	,034	1,000	-,074	,147
	2	,032	,033	1,000	-,075	,139
	3	,058	,041	1,000	-,074	,190
	4	,007	,026	1,000	-,077	,091
	5	,095	,043	,863	-,045	,234
	6	-,016	,025	1,000	-,098	,066
	8	,145*	,039	,010	,019	,270
8	1	-,108	,038	,160	-,231	,015
	2	-,113	,040	,175	-,242	,017
	3	-,087	,049	1,000	-,245	,072
	4	-,138*	,041	,034	-,270	-,005
	5	-,050	,046	1,000	-,200	,100
	6	-,161*	,035	,000	-,274	-,048
	7	-,145*	,039	,010	-,270	-,019

Based on estimated marginal means

a. Adjustment for multiple comparisons: Bonferroni.

*. The mean difference is significant at the ,05 level.

A.4: Comparaciones a pares, considerando el desempeño de los métodos sobre HP2003

(I) algoritmo	(J) algoritmo	Mean Difference (I-J)	Std. Error	Sig.[a]	95% Confidence Interval for Difference[a]	
					Lower Bound	Upper Bound
1	2	,023	,016	1,000	-,028	,074
	3	,062	,022	,165	-,009	,132
	4	,005	,018	1,000	-,051	,062
	5	,038	,022	1,000	-,033	,109
	6	,018	,019	1,000	-,041	,078
	7	,030	,022	1,000	-,040	,101
	8	,274*	,032	,000	,172	,376
2	1	-,023	,016	1,000	-,074	,028
	3	,039	,022	1,000	-,031	,108
	4	-,018	,016	1,000	-,067	,032
	5	,015	,021	1,000	-,052	,082
	6	-,005	,018	1,000	-,060	,051
	7	,007	,019	1,000	-,055	,069
	8	,251*	,032	,000	,150	,353
3	1	-,062	,022	,165	-,132	,009
	2	-,039	,022	1,000	-,108	,031
	4	-,056	,018	,058	-,114	,001
	5	-,024	,020	1,000	-,086	,039
	6	-,043	,020	,851	-,106	,020
	7	-,031	,021	1,000	-,099	,036
	8	,213*	,030	,000	,118	,307
4	1	-,005	,018	1,000	-,062	,051
	2	,018	,016	1,000	-,032	,067
	3	,056	,018	,058	-,001	,114
	5	,033	,017	1,000	-,021	,087
	6	,013	,012	1,000	-,025	,051
	7	,025	,014	1,000	-,019	,070
	8	,269*	,030	,000	,174	,364
5	1	-,038	,022	1,000	-,109	,033
	2	-,015	,021	1,000	-,082	,052
	3	,024	,020	1,000	-,039	,086
	4	-,033	,017	1,000	-,087	,021
	6	-,020	,020	1,000	-,082	,043
	7	-,008	,018	1,000	-,065	,049
	8	,236*	,031	,000	,139	,334
6	1	-,018	,019	1,000	-,078	,041
	2	,005	,018	1,000	-,051	,060
	3	,043	,020	,851	-,020	,106
	4	-,013	,012	1,000	-,051	,025
	5	,020	,020	1,000	-,043	,082
	7	,012	,017	1,000	-,041	,065
	8	,256*	,030	,000	,160	,352
7	1	-,030	,022	1,000	-,101	,040
	2	-,007	,019	1,000	-,069	,055
	3	,031	,021	1,000	-,036	,099
	4	-,025	,014	1,000	-,070	,019
	5	,008	,018	1,000	-,049	,065
	6	-,012	,017	1,000	-,065	,041
	8	,244*	,028	,000	,156	,332
8	1	-,274*	,032	,000	-,376	-,172
	2	-,251*	,032	,000	-,353	-,150
	3	-,213*	,030	,000	-,307	-,118
	4	-,269*	,030	,000	-,364	-,174
	5	-,236*	,031	,000	-,334	-,139
	6	-,256*	,030	,000	-,352	-,160
	7	-,244*	,028	,000	-,332	-,156

Based on estimated marginal means

a. Adjustment for multiple comparisons: Bonferroni.

*. The mean difference is significant at the ,05 level.

A.5: Comparaciones a pares, considerando el desempeño de los métodos sobre HP2004

(I) algoritmo	(J) algoritmo	Mean Difference (I-J)	Std. Error	Sig.[a]	95% Confidence Interval for Difference[a]	
					Lower Bound	Upper Bound
1	2	,047	,018	,372	-,013	,107
	3	,053	,024	,813	-,024	,131
	4	,071	,039	1,000	-,056	,199
	5	,111*	,031	,015	,012	,211
	6	,025	,046	1,000	-,123	,173
	7	,098	,041	,540	-,035	,230
	8	,206*	,046	,001	,058	,354
2	1	-,047	,018	,372	-,107	,013
	3	,007	,030	1,000	-,090	,103
	4	,025	,041	1,000	-,107	,156
	5	,064	,033	1,000	-,044	,173
	6	-,022	,045	1,000	-,168	,124
	7	,051	,041	1,000	-,081	,182
	8	,159*	,047	,029	,008	,310
3	1	-,053	,024	,813	-,131	,024
	2	-,007	,030	1,000	-,103	,090
	4	,018	,041	1,000	-,115	,151
	5	,058	,032	1,000	-,046	,162
	6	-,029	,046	1,000	-,177	,119
	7	,044	,041	1,000	-,088	,176
	8	,152*	,046	,037	,004	,301
4	1	-,071	,039	1,000	-,199	,056
	2	-,025	,041	1,000	-,156	,107
	3	-,018	,041	1,000	-,151	,115
	5	,040	,041	1,000	-,093	,173
	6	-,047	,047	1,000	-,200	,106
	7	,026	,021	1,000	-,042	,094
	8	,134	,042	,051	,000	,269
5	1	-,111*	,031	,015	-,211	-,012
	2	-,064	,033	1,000	-,173	,044
	3	-,058	,032	1,000	-,162	,046
	4	-,040	,041	1,000	-,173	,093
	6	-,086	,043	1,000	-,225	,052
	7	-,014	,040	1,000	-,142	,115
	8	,095	,046	1,000	-,053	,242
6	1	-,025	,046	1,000	-,173	,123
	2	,022	,045	1,000	-,124	,168
	3	,029	,046	1,000	-,119	,177
	4	,047	,047	1,000	-,106	,200
	5	,086	,043	1,000	-,052	,225
	7	,073	,046	1,000	-,075	,221
	8	,181*	,052	,024	,012	,350
7	1	-,098	,041	,540	-,230	,035
	2	-,051	,041	1,000	-,182	,081
	3	-,044	,041	1,000	-,176	,088
	4	-,026	,021	1,000	-,094	,042
	5	,014	,040	1,000	-,115	,142
	6	-,073	,046	1,000	-,221	,075
	8	,108	,043	,376	-,030	,247
8	1	-,206*	,046	,001	-,354	-,058
	2	-,159*	,047	,029	-,310	-,008
	3	-,152*	,046	,037	-,301	-,004
	4	-,134	,042	,051	-,269	,000
	5	-,095	,046	1,000	-,242	,053
	6	-,181*	,052	,024	-,350	-,012
	7	-,108	,043	,376	-,247	,030

Based on estimated marginal means

a. Adjustment for multiple comparisons: Bonferroni.

*. The mean difference is significant at the .05 level.

A.6: Comparaciones a pares, considerando el desempeño de los métodos sobre TD2003

(I) algoritmo	(J) algoritmo	Mean Difference (I-J)	Std. Error	Sig.[a]	95% Confidence Interval for Difference[a]	
					Lower Bound	Upper Bound
1	2	-,008	,021	1,000	-,077	,060
	3	,025	,027	1,000	-,065	,115
	4	-,047	,021	,852	-,117	,023
	5	,001	,024	1,000	-,079	,081
	6	-,031	,020	1,000	-,097	,035
	7	-,034	,027	1,000	-,124	,055
	8	-,013	,018	1,000	-,071	,045
2	1	,008	,021	1,000	-,060	,077
	3	,034	,029	1,000	-,061	,129
	4	-,039	,021	1,000	-,107	,030
	5	,009	,024	1,000	-,068	,087
	6	-,022	,021	1,000	-,090	,046
	7	-,026	,026	1,000	-,113	,061
	8	-,004	,028	1,000	-,096	,088
3	1	-,025	,027	1,000	-,115	,065
	2	-,034	,029	1,000	-,129	,061
	4	-,072	,029	,458	-,168	,024
	5	-,024	,026	1,000	-,109	,061
	6	-,056	,030	1,000	-,154	,043
	7	-,060	,031	1,000	-,161	,041
	8	-,038	,021	1,000	-,106	,030
4	1	,047	,021	,852	-,023	,117
	2	,039	,021	1,000	-,030	,107
	3	,072	,029	,458	-,024	,168
	5	,048	,023	1,000	-,028	,124
	6	,016	,012	1,000	-,025	,058
	7	,013	,025	1,000	-,069	,095
	8	,034	,024	1,000	-,045	,114
5	1	-,001	,024	1,000	-,081	,079
	2	-,009	,024	1,000	-,087	,068
	3	,024	,026	1,000	-,061	,109
	4	-,048	,023	1,000	-,124	,028
	6	-,032	,025	1,000	-,115	,052
	7	-,035	,020	1,000	-,103	,032
	8	-,014	,022	1,000	-,086	,059
6	1	,031	,020	1,000	-,035	,097
	2	,022	,021	1,000	-,046	,090
	3	,056	,030	1,000	-,043	,154
	4	-,016	,012	1,000	-,058	,025
	5	,032	,025	1,000	-,052	,115
	7	-,004	,028	1,000	-,097	,089
	8	,018	,023	1,000	-,059	,095
7	1	,034	,027	1,000	-,055	,124
	2	,026	,026	1,000	-,061	,113
	3	,060	,031	1,000	-,041	,161
	4	-,013	,025	1,000	-,095	,069
	5	,035	,020	1,000	-,032	,103
	6	,004	,028	1,000	-,089	,097
	8	,022	,025	1,000	-,060	,103
8	1	,013	,018	1,000	-,045	,071
	2	,004	,028	1,000	-,088	,096
	3	,038	,021	1,000	-,030	,106
	4	-,034	,024	1,000	-,114	,045
	5	,014	,022	1,000	-,059	,086
	6	-,018	,023	1,000	-,095	,059
	7	-,022	,025	1,000	-,103	,060

Based on estimated marginal means

a. Adjustment for multiple comparisons: Bonferroni.

A.7: Comparaciones a pares, considerando el desempeño de los métodos sobre TD2004

(I) algoritmo	(J) algoritmo	Mean Difference (I-J)	Std. Error	Sig.[a]	95% Confidence Interval for Difference[a]	
					Lower Bound	Upper Bound
1	2	,025	,012	1,000	-,015	,065
	3	-,020	,017	1,000	-,076	,036
	4	-,004	,011	1,000	-,040	,032
	5	-,043	,015	,177	-,092	,006
	6	-,012	,013	1,000	-,055	,032
	7	-,005	,013	1,000	-,047	,037
	8	,011	,013	1,000	-,032	,054
2	1	-,025	,012	1,000	-,065	,015
	3	-,045	,015	,109	-,095	,004
	4	-,030	,011	,307	-,066	,007
	5	-,068*	,015	,001	-,117	-,019
	6	-,037*	,010	,017	-,071	-,004
	7	-,030	,012	,461	-,070	,010
	8	-,014	,016	1,000	-,065	,036
3	1	,020	,017	1,000	-,036	,076
	2	,045	,015	,109	-,004	,095
	4	,016	,015	1,000	-,034	,065
	5	-,023	,014	1,000	-,068	,022
	6	,008	,017	1,000	-,046	,063
	7	,015	,013	1,000	-,026	,056
	8	,031	,014	,989	-,016	,078
4	1	,004	,011	1,000	-,032	,040
	2	,030	,011	,307	-,007	,066
	3	-,016	,015	1,000	-,065	,034
	5	-,038	,012	,055	-,077	,000
	6	-,008	,009	1,000	-,036	,021
	7	-,001	,010	1,000	-,032	,031
	8	,015	,013	1,000	-,026	,057
5	1	,043	,015	,177	-,006	,092
	2	,068*	,015	,001	,019	,117
	3	,023	,014	1,000	-,022	,068
	4	,038	,012	,055	,000	,077
	6	,031	,014	1,000	-,016	,077
	7	,038*	,011	,017	,004	,072
	8	,054*	,015	,013	,006	,101
6	1	,012	,013	1,000	-,032	,055
	2	,037*	,010	,017	,004	,071
	3	-,008	,017	1,000	-,063	,046
	4	,008	,009	1,000	-,021	,036
	5	-,031	,014	1,000	-,077	,016
	7	,007	,012	1,000	-,032	,046
	8	,023	,015	1,000	-,026	,072
7	1	,005	,013	1,000	-,037	,047
	2	,030	,012	,461	-,010	,070
	3	-,015	,013	1,000	-,056	,026
	4	,001	,010	1,000	-,031	,032
	5	-,038*	,011	,017	-,072	-,004
	6	-,007	,012	1,000	-,046	,032
	8	,016	,011	1,000	-,020	,052
8	1	-,011	,013	1,000	-,054	,032
	2	,014	,016	1,000	-,036	,065
	3	-,031	,014	,989	-,078	,016
	4	-,015	,013	1,000	-,057	,026
	5	-,054*	,015	,013	-,101	-,006
	6	-,023	,015	1,000	-,072	,026
	7	-,016	,011	1,000	-,052	,020

Based on estimated marginal means

a. Adjustment for multiple comparisons: Bonferroni.

*. The mean difference is significant at the ,05 level.