Estado del arte en Probabilistic Latent Semantic Analysis aplicado a problemas de acceso a la información en la Web.

José Alberto Benítez Andrades

Octubre 2011

En este trabajo se resumen las conclusiones obtenidas después de haber realizado la lectura de los artículos propuestos de Sergy Brin amf Lawrence Page, Junghoo Cho, Héctor García-Molina, además de los que tratan sobre el rastreador Mercator, escritos por Allan Heydon y Marc Najork.

1. Introducción.

1.1. Qué es "Probabilistic Latent Semantic Analysis" (PLSA).

El Análisis Probabilístico Semántica Latente (PLSA), también conocido como indexación semántica latente probabilística (PLSI, especialmente en los círculos de recuperación de información) es una técnica estadística para el análisis de dos modos y los datos de co-ocurrencia. PLSA ha evolucionado desde el análisis semántico latente, la adición de un modelo probabilístico lo hizo más sólido.

PLSA tiene aplicaciones en la recuperación de la información y filtrado, procesamiento del lenguaje natural, aprendizaje automático a partir del texto, y áreas relacionadas. Fue introducido en 1999 por **Jan Puzicha** y **Thomas Hofmann**, y se relaciona con la factorización de matrices no negativas.

En comparación con el análisis semántico latente estándar que se deriva de álgebra lineal y downsizes las tablas de ocurrencia (por lo general a través de una descomposición de valor singular), el análisis semántico latente probabilístico se basa en una mezcla de derivados de la descomposición de un modelo de clases latentes. Esto da como resultado un enfoque más de principios que tiene una sólida base en las estadísticas. Teniendo en cuenta las observaciones en forma de co-ocurrencias (w, d) de las palabras y documentos, en los modelos PLSA la probabilidad de cada co-ocurrencia se percibe como una mezcla de distribuciones multinomial condicionalmente independientes.

1.2. ¿ En qué consiste ?

Los modelos probabilistas parten de distribuciones de probabilidad para representar el conocimiento encerrado en el lenguaje. En general, estos modelos están diseñados para aplicaciones específicas, típicamente para clasificación de textos y recuperación de información, puesto que el conocimiento que son capaces de recoger es limitado.

El modelo PLSA fue presentado y aplicado por primera vez en la minería de texto por Hoffman. En contraste con el algoritmo estándard LSI, que utiliza la norma de Frobenius como un criterio de optimización, el modelo PLSA se basa en el principio de probabilidad máxima, que es derivado de teorías estadísticas dudosas. Básicamente, el modelo PLSA se basa en el modelo estadístico llamado modelo de aspecto, que puede ser utilizado para identificar relaciones semánticas ocultas mediante actividades de co-ocurrencia.

Teóricamente, podemos ver las sesiones de los usuarios sobre las páginas web como actividades co-ocurrentes en el contexto de la minería de uso web, para concluir el patrón de uso latente. Dado el modelo de aspecto sobre el patrón de acceso de usuario en el contexto de la minería web, primero se asume que hay un factor latente de espacio Z = (z1, z2, $\cdots$ zk), y cada lista de datos derivada de la observación de co-ocurrencia (s_i, p_j) (por ejemplo, la visita de la página p_j en la sesión de usuario s_i) es asociada con el factor z_k. Acorde a este punto de vista, el modelo de aspecto puede concluir en que existen diferentes relaciones entre los usuarios web o las páginas correspondientes a diferentes factores. Además, los diferentes factores pueden ser considerados para representar los correspondientes patrones de acceso a usuario. Por ejemplo, durante un proceso de minería de uso web en una tienda online, nosotros podemos definir que existen k factores latentes asociados con k tipos de patrones de navegación, como . Furthermore, the different factors can be considered to represent the corresponding user access pattern. For example, during a Web usage mining process on an e-commerce website, we can define that there exist k latent factors associated with k kinds of navigational behavior patterns, así como el factor z_1 para los que tienen interest en los productos deportivos, z_2 para productos con interés en venta and y z_3 para buscar a través de la variedad de páginas de productos en diferentes categorías.

De esta manera, cada uno de los datos de observación de co-ocurrencia (s_i, p_j) puede transmitir interés de los usuarios de navegación mediante la asignación de los datos de observación en el espacio latente k-dimensional de los factores. El grado, a la que este tipo de relaciones se "explican" por cada uno de los factores, se obtiene una distribución de probabilidad condicional asociada con los datos de uso de la Web. Por lo tanto, el objetivo de emplear el modelo PLSA, es determinar la distribución de probabilidad condicional, a su vez, para revelar las relaciones intrínsecas entre los usuarios de la Web o las páginas basadas en un enfoque de inferencia de probabilidad. En una palabra, el modelo PLSA es modelo y el comportamiento de navegación del usuario puede concluir en un espacio semántico latente, e identificar el factor latente asociado. Antes de proponer el algoritmo basado PLSA para la minería uso de la Web, es necesario

introducir la formación matemática del modelo PLSA, y el algoritmo que se utiliza para estimar la distribución de probabilidad condicional.

- P(s_i) representa la probabilidad que será observada en una sesión de usuario particular s_i,
- P(z_k | s_i) representa una probabilidad de una sesión de usuario específica sobre la clase z_k factor latente,
- P(p_j | z_k) representa la probabilidad de distribución de la clase condicional de las páginas sobre una variable latente z_k.

Basándonos en las siguientes definiciones, el modelo PLSA puede expresarse de la siguiente forma:

- Seleccionando una sesión de usuario s_i con una probabilidad de P(s_i),
- Escogiendo un factor oculto $z_k$con una probabilidad P(z_q | s_i),
- Generando una página $p_j$con una probabilidad P(p_j | z_k) ;

Y como resultado, obtendríamos una probabilidad ocurrente de un par observado (z_q | p_j) que adopta el factor variable z_k. Traduciendo este proceso en un modelo de probabilidad de resultados en la expresión:

P(s_i | p_j) = P(s_i) · P(p_j |s_i)

donde P(p_j ,s_i) = $\sum_{z \epsilon Z}$P(p_j | z) · P(z |s_i)

Aplicando la fómrula Bayesiana, una versión reparametrizada puede ser transformada en la ecuación siguiente:

P(p_j,s_i) = $\sum_{z \epsilon Z}$P(z) P(s_i | z) P(p_j | z)

Siguiendo el principio de probabilidad, nosotros podemos determinar la probabilida total de la observación como:

asasasasas

donde $m(s_i, p_j)$ corresponde a la entrada de la matriz asociada de la sesión de páginas vistas con la sesión s_i y la página vista p_j. Para mazimizar la probabilidad total, es necesario generar repetidamente las probabilidades condicionales de P(z) , P(s_i | z) y P(p_j | z) utilizando el uso de los datos de observación. Sabiendo las estadísticas, el algoritmo de EM (Expectation-Maximization) es un procedimiento eficiente para mejorar la estimación de probabilidad máxima en el modelo latente variable. Generalmente se necesitan dos pasos para implementar en el procedimiento alternativamente: (1) el paso de "Expectation" (E), donde las probabilidades posteriores son calculadas por los factores latentes basados en las estimaciones actuales de la probabilidad condicional, y la Maximización (Maximization (M)), que es el paso donde las probabilidades estimadas condicionales se actualizan e intentan maximizar la probabilidad basadas en las probabilidades posteriores computadas en el paso anterior.

Todo el procedimiento se da como sigue: En primer lugar, dados los valores al azar inicial de P (z), P (si | z), P (pj | z), entonces, en el paso E, podemos simplemente aplicar la fórmula bayesiana para generar el siguiente variable basada en la observación de su uso:

además, en el paso de Maximización, se computa:

asassasaasas

P(si|zk) = f° pj¸¸P m(si, pj)P(zk|si, pj) f° s i ¸S,pj¸P m(s i, pj)P(zk|s i, pj) (6.18)

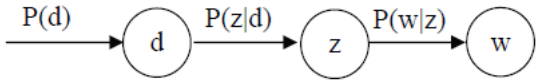
P(zk) = 1 R f° si¸S,pj¸P m(si, pj)P(zk|si, pj) (6.19)

where R = f° si¸S,pj¸P m(si„ pj) q

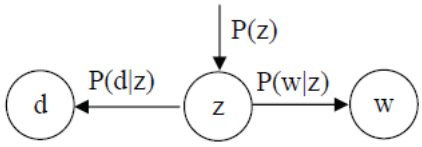
Básicamente, susistituyendo las ecuaciones (6.17)-(6.19) en (6.14) and (6.15) resultará en el que se incrementa la probabilidad de Li total de los datos de observación. La implementación iterativa de la E-paso y paso-M se repite hasta que Li está convergiendo hacia un límite de locales óptimos, lo que significa que los resultados calculados pueden representar las estimaciones de probabilidad óptima de los datos de uso de la observación. De la formulación anterior, es fácil de encontrar que la complejidad computacional del modelo PLSA es O (MNK), donde m, n, k denota el número de sesiones de usuario, páginas Web y los factores latentes, respectivamente.

Por ahora, hemos obtenido la distribución de probabilidad condicional de P (ZK), P (si | zk) y P (pj | zk) mediante la realización de la E y el paso M iterativa. La distribución de probabilidad que se estima que es correspondiente a la máxima verosimilitud local contiene la información útil para inferir el uso de factores semánticos, el usuario performing Web sesiones de la agrupación que se describen en las secciones siguientes.

El modelo de Análisis de Semántica Latente Probabilista PLSA (Probabilistic Latent Semantic Analysis) también podemos considerar que propone un modelo de probabilidad, dados un documento y una palabra, de dos maneras distintas, como muestra la figura 1.2.1 (es otra forma de representarlo) . En el caso a) el modelo es asimétrico y en el caso b) simétrico, siendo d el documento, q la palabra, z la categoría o tópico y P la función de probabilidad.



a)



b)

Al igual que en LSA, el experto ha de definir clases, tópicos o pasajes en los que incluir los textos y palabras. Así pues, los modelos de la figura responden a las expresiones siguientes:

a) P(d,w) = $\sum_z P(w|z) \cdot P(z|d)$

b) P(d,w) = $\sum_z P(z) \cdot P(d|z) \cdot P(w|z)$

Estos modelos son posteriormente ajustados mediante un algoritmo EM (*Expectation Maximization*), obteniendo así tres matrices: U, que contiene las probabilidades de los documentos dadas las categorías, V, que contiene las probabilidades de las palabras dadas las categorías y la matriz diagonal $\sum$, que contiene las probabilidades de las categorías. El modelo queda definido finalmente como $P = U \bullet V \bullet \sum$.

Aunque se pueda establecer cierta analogía con la descomposición en valores singulares de LSA, la matriz P presenta ciertas diferencias ventajosas:

- P es una distribución de probabilidad bien definida y los factores tienen un claro significado probabilista, contrariamente a LSA.
- Las direcciones de los vectores en el espacio LSA no tienen interpretación. En PLSA son distribuciones multinomiales de palabras.
- La elección del número de dimensiones del espacio tiene un trasfondo teórico en PLSA. En LSA se hace de manera experimental.

Experimentos realizados en recuperación de información demuestran que PLSA obtiene una precisión entrono al 10 % mejor que LSA en varios corpus de textos.

El modelo de Localización de Dirichlet Latente LDA proporciona, según sus autores, una semántica completa probabilística generativa para documentos. Los documentos se modelan mediante una variable aleatoria oculta de Dirichlet. Al igual que en PLSA, asume un conjunto de categorías predefinidas. Cada categoría se representa como una distribución multinomial sobre el conjunto de palabras del vocabulario. El modelo queda descrito por la siguiente expresión:

$$\mathrm{P(d)} = \int \vartheta \; [\; \Pi \mathrm{n} \; \Sigma \mathrm{z} \; \mathrm{P(wn|zn;\beta)} \cdot \mathrm{P(zn|\vartheta)} \;] \cdot \mathrm{P(\vartheta;\alpha)} \; \delta\vartheta$$

siendo $\mathrm{P}(\vartheta;\alpha)$ una distribución de Dirichlet, $\mathrm{P(zn|\vartheta)}$ una multinomial que indica el grado en que el tema zn es tratado en un documento y β una matriz de clases por palabras del vocabulario. De esta manera, la probabilidad de un documento depende de las probabilidades de que sus palabras denoten ciertas categorías dentro de una distribución de Dirichlet. Para aprender e inferir en el modelo usan un algoritmo EM análogo al modelo PLSA descrito anteriormente.

El modelo de Mezcla de Unigramas [Nigam et al., 2000] es un modelo sencillo y muy similar a los dos sistemas anteriores. Está descrito por la siguiente expresión:

$$\mathrm{P(d)} = \Sigma \mathrm{z} \; (\; \Pi \mathrm{n} \; \mathrm{P(wn|z)} \;) \cdot \mathrm{P(z)}$$

Como se aprecia, la probabilidad de un documento depende de las probabilidades de que sus palabras pertenezcan a las categorías.

Experimentos en clasificación de textos y en recuperación de información muestran la superioridad del modelo LDA con respecto a PLSA y al modelo de Mezcla de Unigramas. Además, LDA recoge la posibilidad de que un documento contenga más de una categoría temática, al contrario que la Mezcla de Unigramas, y no está condicionado por los ejemplos de entrenamiento, como es el caso de PLSA.

1.3. Aplicaciones del PLSA en distintos estudios.

Este método probabilístico latente, desde su creación en 1999 por Hoffman, ha sido utilizado con diferentes fines y modificado por disintos investigadores, con la intención de mejorar su funcionamiento aplicando distintos criterios.

Leyendo algunos de los proceedings de los últimos años, he seleccionado cinco estudios que me han parecido bastante interesantes, en los que se aplica la técnica de PLSA, con distintas modificaciones para poder conseguir una mejora en los resultados de los experimentos realizados.

Los proceedings que he elegido son los siguientes:

- Adaptive Label-Driven Scaling for Latent Semantic Indexing. SIGIR 2010
- Topic-bridged PLSA for Cross-Domaint Text Classification. SIGIR 2010
- ILDA: Interdependent LDA Model for Learning Latent Aspects and their Ratings from Online Product Reviews. SIGIR 2011.
- Clickthrough-Based Latent Semantic Models for Web Search. SIGIR 2011.
- Regularized Latent Semantic Indexing. SIGIR 2011.

1.3.1. Adaptive Label-Driven Scaling for Latent Semantic Indexing.

En primer lugar, comentar que los autores de este proceeding son *Xiaojun Quan, Enhong Chen, Quiming Luo y Hui Xiong.* Pertenecen al departamento de ciencias de la computación de la Universidad de Ciencia y Tecnología de China (USTC).

Este trabajo de investigación, se trata principalmente de mejorar la técnica de LSI mediante la explotación de etiquetas de categoría. Específicamente, en la matriz de términos de documento, el vector para cada término apareciendo en etiquetas o semánticamente cercano a las etiquetas, se escala antes de realizar la técnica de SVD (Singular Value Descomposition [Descomposición de valor singular]) para aumentar su impacto en la generación de vectores singulares. Como resultado, las similitudes entre los documentos pertenecientes a una misma categoría, se incrementan. Además, se diseña una estrategia de escalado adaptativo para mejorar la utilización de estructuras de herencia para las categorías. Los resultados de este experimento muestran que el enfoque que proponen sus autores, mejora significativamente la actuación de la categorización de texto por herencia.

En su proceeding, destacan en la **introducción** que la Indexación Semántica Latente (LSI), es una técnica de recuperación y categorización de texto que cuenta con diferentes frameworks para poder aplicarlo y que ha recibido anteriormente distintas mejoras en otros estudios realizados por otros investigadores. Ellos proponen un enfoque de aplicación del LSI explotando las etiquetas de categoría, como indiqué anteriormente.

En la **metodología**, explican que en la categorización de texto, los términos en un documento que también aparecen en etiquetas de categorías son más efectivos categorizando el documento que otros términos. Se encargaron de estudiar

una estrategia para impulsar el impacto. Su propuesta fue escalar los vectores de términos de etiquetas de categoría en la matriz de términos del documentos antes de impulsar el SVD.

Extienden el hecho de escalar una serie de términos que son similares a las etiquetas. Estos términos aparecen en las etiqeutas o son similares a las etiquetas de categorías. Estos términos se llaman "label-relevant" (etiquetas relevantes). Estos términos se basan en la siguiente fórmula:

$$\text{label-relevant (t)} = \{s|\text{rank}(\text{sim}(s,t)) \leq l\}$$

En esta fórmula sim(s,t) representa la similitud entre s y t. Demuestran que mediante su método "label-driven scaling", se incrementa la similitud de una consulta con un documento de la misma categoría. Desarrollan su estudio de forma matemática y sacan una serie de conclusiones.

Explican, que en uno de los ejemplos, cuando la consulta y el documento pertenecen a la misma categoría, ellos tienen más probabilidades de tener un término "label-relevant".

Y en relación a la categorización de texto por herencia, explican que la organización de las categorías es mediante herencia, y que las que se encuentran en el nivel más inferior del árbol de herencia, son las más especificas.

Los experimentos o pruebas que realizan, consisten en colecciones de datos que tienen ya las categorías organizadas como taxonomías y cuya etiqueta para cada categoría está predefinida. Estos documentos son preprocesados siempre. Después de eliminar una "stopword", ellos se encargan de filtrar términos con menos de dos caracteres. Para decidir qué términos son "label-relevant", ellos utilizan LSI con la reducción de dimensión en 50. En el proceso de clasificación, el número de vecinos cercanos, se configura en 20. Finalmente, ellos comparan la actuación de dos variantes de su enfoque con dos enfoces: clasificadores kNN cuyas similitudes se obtienen en el espacio LSI; y clasificadores SVM de herencia, usando un núcleo lineal y unos parámetros con valores por defecto. La diferencia entre NADP y SLSI es que el formador aplica un escalado uniforme a todos los nodos en la herencia mientras el último aplica un escalado de adaptación. En muchos casos, para el escalado "label-driven" y para el escalado de adaptación, se mejora la clasificación de la actuación.

Como conclusiones, declaran que sus dos enfoques del LSI: escalado "label-driven" y de adaptación, mejoran los resultados en datos del mundo real, gracias a la categorízación con herencia.

1.3.2. Topic-bridged PLSA for Cross-Domain Text Classification.

Este trabajo de investigación fue realizado por *Gui-Rong Xue, Wenyuan Dai, Qiang Yang y Yong Yu.* Pertenecen a la *Universidad de Ciencia y Tecnología Clearwater Bay, Knowloon, Hong Kong.*

En muchas aplicaciones web, como por ejemplo la clasificación de blogs, clasificación de grupos de noticias, o datos etiquetados, son escasos. Obtener etiquetas de un nuevo dominio, suele ser caro y consume mucho tiempo, mientras que puede haber un conjunto de datos etiquetado en un dominio distinto

pero que se relaciona con el nuevo. Los métodos de clasificación de texto antiguos no permiten aprender cruzando distintos dominios. Estos investigadores proponen un algoritmo de clasificación de texto mediante el cruce de dominios que en definitiva, extiende el PLSA para integrar datos etiquetados y datos no etiquetados que vienen de dominios distintos pero que están relacionados, en un modelo probabilístico unificado. A este algoritmo nuevo lo llaman Topic-bridged PLSA (TPLSA). El algoritmo consiste en explotar los temas entre dos dominios y transferir la base de conocimiento entre esos dominios mediante un "puente de temas" (topic-bridge), que ayuda a la clasificación de texto en el dominio de destino. Una ventaja única que tiene su método es la capacidad para extraer al máximo el conocimiento que luego puede ser transferido entre los dominios. Esto hace que este algoritmo sea de los mejores en cuanto a clasificación de texto se refiere.

Explican primeramente cuáles son las tareas que realizan los framework en el aprendizaje tradicional. Dan mucha importancia y recalcan varias veces que etiquetar nuevos dominios es costoso en cuanto a tiempo sobre todo. Hay que tener en cuenta también, que en una web que se actualiza con mucha frecuencia, es complejo tener todas las etiquetas actualizadas, si no se consigue de manera automática.

Lo que ellos proponen parte de dos conjuntos de datos D_L y D_U, que están relacionados pero pertenecen a distintos dominios. D_L representa el conjunto de datos etiquetados del dominio antiguo y D_U pertence al nuevo dominio y necesita ser clasificado. Las etiquetas que pertenecen a D_L y las que van a predecirse para D_U son creadas desde el mismo conjunto de etiquetas C. El objetivo es clasificar al completo el conjunto Du a través del dominio antiguo y su conjunto de datos D_L.

La principal ventaja de este algoritmo es que extendiendo el modelo PLSA para datos desde distintos dominios (los de entrenamiento y los de pruebas), permiten indicar partes de la base de conocimiento a través del TPLSA que son constantes entre diferentes dominios y partes que son específicas de cada uno. Esto permite transferir la base de conocimiento aprendida incluso cuando los dominios son diferentes.

En este proceeding, explican que existen distintos tipos de clasificadores de texto tradicionales, con aprendizaje supervisado y semi-supervisado, pero que no sirven para el experimento que desean hacer funcionar, la transferencia entre dominios distintos.

TPLSA: Definición del problema, aprendizaje y pruebas. Los elementos principales para aplicar este algoritmo son: documento d que representa la instancia entrenada, que asu veces es asignada a una etiqueta única desde un conjunto temático $C = \{c1, \ldots, ck\}$. Un vocabulario de palabras $W = \{w1,\ldots,wv\}$ que es dado y representa una bolsa de palabras. A partir de estos elementos, se trabaja con los conjuntos de documentos etiquetados y sin etiquetar D_L y D_U.

Este modelo TPLSA se puede dividir en dos partes. En relación con el con-

junto de documentos etiquetados del dominio antiguo, podemos decir que PLSA actúa en D_L x W de la siguiente manera:

$Pr(d_l|w) = \sum Pr(d_l|z)Pr(z|w)$

donde $d_l \epsilon D_L$es el documento del conjunto entrenado.

Para el conjunto de datos de prueba D_U acorde con la observación de D_U y W, podemos decir que D_U x W se define de la siguiente forma:

$Pr(d_u|w) = \sum Pr(d_u|z)Pr(z|w)$

donde $d_u \epsilon D_U$es el documento del conjunto de pruebas.

Teniendo en cuenta esto, mediante una serie de ecuaciones, se relacionan las probabilidades condicionales de los documentos de prueba y los documentos ya entrenados y se van creando las etiquetas para los documentos del nuevo dominio.

Una vez hecho esto, se optimiza mediante otra serie de ecuaciones de probabilidad y lógica y se aplica el algoritmo de EM para asegurar que el valor de las funciones cumplen la optimización.

Las pruebas que realizan parten de la base de 3 conjuntos de datos.

Conclusiones finales del TPLSA. Después de realizar la evaluación de 11 conjuntos de datos, los resultados que obtienen estos investigadores son muy positivos, ya que muestran que el algoritmo propuesto logra mejorar la actuación con respecto a otros algoritmos de clasificación.

En un futuro, considerarán otros métodos de aprendizaje para adquirir los parámetros usados en el modelo TPLSA y considerar otroas tareas de clasificación relacionadas como la clasificación "multi-clase".

En definitiva, la mejora que le añaden al modelo PLSA natural, es la inclusión de dos conjuntos de datos de distintos dominios que pueden transferirse entre ellos las categorías de sus términos.

1.3.3. ILDA: Interdependent LDA Model for Learning Latent Aspects and their Ratings from Online Product Reviews.

Esta investigación fue realizada por *Samaneh Moghaddam y Martin Ester*, ambas pertenecen a la *School of Computing Science, Simon Fraser University, Burnaby, BC, Canada.*

Hoy en día hay muchísimos análisis y críticas de distintos productos en Internet, de hecho, existen foros especializados para analizar distintos tipos de productos, blogs y grupos de discusión. Un ejemplo de ello puede ser Xataka, que analiza los distintos teléfonos móviles que salen al mercado.

Sin embargo, para una persona que desea comprar un producto, es bastante complejo en muchas ocasiones, poder llegar a contrastar todos los análisis que hay sobre un mismo producto en distintas webs. Esta dificultad ha provocado que la minería de opinión haya creado una nueva rama de investigación.

Los análisis de productos se componen de distintos elementos: por ejemplo, el aspecto es un atributo de un producto, en el caso de una cámara, la pantalla es un atributo de la misma. Teniendo en cuenta distintos elementos, los usuarios

suelen puntuar los distintos componentes del producto y al finalizar, se hace una media de todas las puntuaciones y esa es la puntuación que obtiene el producto.

Este equipo de investigación propone tres modelos probabilísticos gráficos que extraen los distintos aspectos y puntuaciones de los productos de análisis que hay en la red. Los primeros dos modelos extienden el standard PLSI y LDA para generar un resumen de puntuación de aspecto. Introducen el modelo ILDA (Interdependent Latent Dirichlet Allocation). Las pruebas que realizan en sus experimentos, utilizan conjuntos de datos reales, obtenidos de Epinions.com y demuestran la mejora en la efectividad del modelo ILDA.

Trabajos relacionados y definición del problema. En este proceeding se explica que los trabajos realizados hasta el momento relacionados con la minería de opinión, en su gran mayoría, han sido enfocados a la tarea de identificación, y han ignorado el problema de la predicción de puntuación. Aún así, existen trabajos enfocados a este problema, que intentan solucionarlo basándose en bolsas de palabras, recuperando distintas líneas de los análisis que hay en Internet.

Lo que proponen estos es utilizar dos modelos de asociación latente semántica, el primero agrupa palabras en un conjunto de aspectos acorde con el contexto y el segundo agrupa palabras acorde a sus estructuras latentes semánticas y al contexto de los disintos análisis que hay de cada producto.

Otros investigadores, presentan modelos probablísticos gráficos para el modelado de contenido de páginas independientes y de capas de información de páginas dependientes. Pero también ignoran las puntuaciones de estos análisis.

En definitiva, todas las ramas de investigación que emergían de este problema, iban enfocadas al mismo problema, pero no resolvían el problema que en este caso manejan.

Los investigadores de este proyecto, parten de un conjunto P = {P1,P2, ... , Pl} que representa un conjunto de los productos que pueden ser de categorías distintas. Para cada producto Pi hay un conjunto de análisis del mismo Ri = {d1, d2,...,dN}. Cada análisis dj consiste en un conjunto de frases de opinión como por ejemplo "gran zoom", "excelente calidad", etc. Se puede decir que el problema se descompone en los siguientes elementos:

- Aspecto: El aspecto es un atributo o componente de un producto que ha sido comentado en un análisis. Por ejemplo, "duración de la batería", en la frase "La duración de la batería de esta cámara es bastante corta".
- Puntuación: La puntuación es la satisfacción de un usuario con términos numéricos. La mayoría de las webs disponen de un sistema que va desde 1 hasta 5.
- Frase de opinión: Una frase de opinión f = < t, s > es un par de términos t y sentimientos s. Normalmente el término es un aspecto y el sentimiento expresa la opinión, por ejemplo: < duración de la batería, corta >.
- Análisis: El análisis es una bolsa de frases de opinión.

- Definición del problema: Dado un conjunto de análisis para un producto P, la tarea es identificar los k principales aspectos de P y entonces predecir la puntuación de cada aspecto.

Teniendo en cuenta esto, podemos tener la valoración de distintos aspectos de un teléfono móvil, en cual, se evalúan por ejemplo la cámara, las dimensiones, la calidad de grabación de vídeo, el sistema operativo y la velocidad del procesador. Teniendo en cuenta estos valores, se hace una media total de cada móvil y sale una puntuación final. Pero se puede dar el caso, lógicamente, de que un móvil A tenga mayor puntuación media que un móvil B, pero si lo que buscamos es que tenga una cámara con mayor calidad, es posible que la B posea esta característica.

El objetivo de este proyecto de investigación es encontrar o predecir las distintas puntuaciones de estos "aspectos" analizando los distintos análisis de un mismo producto que se encuentran por internet, de forma automática.

Valor del PLSI en esta investigación. Como comenté anteriormente, estos investigadores realizan pruebas utilizando tres modelos probabilísticos distintos. Yo voy a resumir únicamente la parte en la que se habla del PLSI.

El PLSI ha sido aplicado a distintos problemas de mineria de texto recientemente. Sin embargo, usaban solo esto para la identificación de aspectos y estos modelos no generaban una puntuación de los productos. Para ello, en este estudio, extienden el modelo del PLSI para identificar aspectos y predecir las puntuaciones, de forma simultanea. Siguiendo el modelo estándard gráfico, los nodos representan variables aleatorias y ejes indicando posibles dependencias. Los nodos "con sombra" son variables aleatorias observadas y los nodos "sin sombra" son variables aleatorias latentes. La parte exterior representa análisis o criticas y la parte interior representa opiniones. N y M son el número de críticas o análisis de productos y el número de opiniones en cada crítica, respectivamente. Si M es independiente de todas las otras variables de datos generadas (a y r), esto se ignora.

Para extender el PLSI en esta investigación, añadieron una segunda fila. Para cada producto P, se genera un modelo de PLSI para asociar un aspecto no observado am y una puntuación rm con cada observación, por ejemplo, cada frase de opinión f = $< t_m, s_m >$ en una crítica $d \epsilon R$. Se puede definir el modelo PLSI adaptado generativo de la siguiente forma:

1. Seleccionar una crítica d de R con probabilidad *P(d)*.
2. Para cada frase de opinión $< t_m, s_m >$, $m \epsilon \{1, 2, ..., M\}$
   - *a*) Ejemplo $a_m \sim P(a_m |\ d)$ y $r_m \sim P(r_m |\ d)$.
   - *b*) Ejemplo $t_m \sim P(t_m |\ a_m)$ y $P(s_m |\ r_m)$.

Traduciendo este proceso en una distribución de probabilidad, la expresión obtenida es la siguiente:

FORMULA 1

Realizando una serie de modificaciones, la fórmula final para la utilización de este modelo es la siguiente:
FORMULA2

Conclusiones de la investigación Resumir los aspectos evaluados da una información muy útil a los usuarios que van a realizar uan compra. La propuesta de estos investigadores es un modelo que enseña un conjunto de aspectos de productos y sus correspondientes puntuaciones de una colección de críticas del producto que han sido preprocesadas en una colección de frases de opinión. Reconocen cuáles son los aspectos más importantes mediante su modelo ILDA y así realizar una evaluación más coherente.

Una de las desventajas de los modelos no supervisados es que la correspondencia entre agrupaciones generadas y variables latentes no son explícitas. Planean extender su trabajo, planean investigar la corresponencia entre las agrupaciones identificadas y los aspectos reales o puntuaciones.

2. En qué conferencias internaciçonales se aborda el crawling.

Algunas de las conferencias internacionales que abordan el tema del crawling, son las siguientes:

- Conferences on World Wide Web.
- ACM International Conference on Management of Data (SIGMOD)
- International Conference on Very Large Databases
- IEEE International Conference on Data Engineering
- European Conference on Research and Advanced Technology for Digital Libraries
- Joint Conference on Digital Libraries (JCDL)
- International Conference on Visual Information Systems (Visual99)
- Latin American Web conference
- International Conference on Machine Learning (ICML97)
- International Conference on Autonomous Agents (Agents '98)
- International Conference on Web Information Systems and Technologies

Referencias

[1] Kobayashi, M. and Takeda, K. (2000). "Information retrieval on the web". ACM Computing Surveys (ACM Press) 32 (2): 144–173.

[2] Sergey Brin amf Lawrence Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. Computer Networks and ISDN Systems, vol. 30, 1998.

[3] Junghoo Cho, Hector Garcia-Molina, Lawrence Page. Efficient Crawling Through URL Ordering. Computer Networks and ISDN Systems, vol. 30, 1998.

[4] Allan Heydon and Marc Najork. Mercator: A Scalable, Extensible Web Crawler. In Proceedings of World Wide Web, 1999, pages 219-229.

[5] Brian Pinkerton. Finding what people want: Experiences with the WebCrawler. In Proc. 1st International World Wide Web Conference, 1994.

[6] Castillo, Carlos (2004). Effective Web Crawling. (Ph.D. thesis). University of Chile. Retrieved 2010-08-03.

[7] Gulli, A.; Signorini, A. (2005). "The indexable web is more than 11.5 billion pages". Special interest tracks and posters of the 14th international conference on World Wide Web. ACM Press.. pp. 902–903.

[8] Baeza-Yates, R., Castillo, C., Marin, M. and Rodriguez, A. (2005). Crawling a Country: Better Strategies than Breadth-First for Web Page Ordering. In Proceedings of the Industrial and Practical Experience track of the 14th conference on World Wide Web, pages 864–872, Chiba, Japan. ACM Press.

[9] Menczer, F. (1997). ARACHNID: Adaptive Retrieval Agents Choosing Heuristic Neighborhoods for Information Discovery. In D. Fisher, ed., Machine Learning: Proceedings of the 14th International Conference (ICML97). Morgan Kaufmann

[10] Diligenti, M., Coetzee, F., Lawrence, S., Giles, C. L., and Gori, M. (2000). Focused crawling using context graphs. In Proceedings of 26th International Conference on Very Large Databases (VLDB), pages 527-534, Cairo, Egypt.

[11] Pant, Gautam; Srinivasan, Padmini; Menczer, Filippo (2004). "Crawling the Web". In Levene, Mark; Poulovassilis, Alexandra. Web Dynamics: Adapting to Change in Content, Size, Topology and Use. Springer. pp. 153–178.

[12] Cho, Junghoo; Hector Garcia-Molina (2003). "Estimating frequency of change". ACM Trans. Interet Technol. 3 (3): 256–290.

[13] Baeza-Yates, R. and Castillo, C. (2002). Balancing volume, quality and freshness in Web crawling. In Soft Computing Systems – Design, Management and Applications, pages 565–572, Santiago, Chile. IOS Press Amsterdam.

[14] Risvik, K. M. and Michelsen, R. (2002). Search Engines and Web Dynamics. Computer Networks, vol. 39, pp. 289–302, June 2002.

[15] Zeinalipour-Yazti, D. and Dikaiakos, M. D. (2002). Design and implementation of a distributed crawler and filtering processor. In Proceedings of the Fifth Next Generation Information Technologies and Systems (NGITS), volume 2382 of Lecture Notes in Computer Science, pages 58–74, Caesarea, Israel. Springer.

[16] Dill, S., Kumar, R., Mccurley, K. S., Rajagopalan, S., Sivakumar, D., and Tomkins, A. (2002). Self-similarity in the web. ACM Trans. Inter. Tech., 2(3):205–223.

[17] Abiteboul, Serge; Mihai Preda, Gregory Cobena (2003). "Adaptive on-line page importance computation". Proceedings of the 12th international conference on World Wide Web. Budapest, Hungary: ACM. pp. 280–290. doi:10.1145/775152.775192. ISBN 1-58113-680-3.

[18] Jon Kleinberg, Authoritative Sources in a Hyperlinked Environment, Proc. ACM-SIAM Symposium on Discrete Algorithms, 1998.

[19] Massimo Marchiori. The Quest for Correct Information on the Web: Hyper Search Engines. The Sixth International WWW Conference (WWW 97). Santa Clara, USA, April 7-11, 1997.

[20] Marc Najork and Janet L. Wiener. Breadth-first crawling yields high-quality pages. In Proceedings of the Tenth Conference on World Wide Web, pages 114–118, Hong Kong, May 2001. Elsevier Science.

[21] Web de Princeton: http://www.cs.princeton.edu/courses/archive/spring10/cos435/Notes/web_c

[22] Archive.org : http://crawler.archive.org/faq.html

[23] CS.CMU.EDU: http://www.cs.cmu.edu/~rcm/websphinx/

[24] Parc.com : http://www2.parc.com/spl/projects/modrobots/chain/polybot/index.html

[25] Crawling the web: http://dollar.biz.uiowa.edu/~pant/Papers/crawling.pdf

[26] Escribiendo un rastreador web en JAVA : http://java.sun.com/developer/technicalArticles/ThirdParty/WebCrawler/