Aproximaciones a SVM semisupervisado multiclase para clasificación de páginas web

Arkaitz Zubiaga Mendialdua

Departamento de Lenguajes y Sistemas Informáticos

ETS. de Ingeniería Informática - UNED

Septiembre de 2008

Departamento de Lenguajes y Sistemas Informáticos

ETS. de Ingeniería Informática

Universidad Nacional de Educación a Distancia

Memoria de Tesis de Máster

Máster en Tecnologías del Lenguaje en la Web

(curso 2007-2008)

Aproximaciones a SVM semisupervisado multiclase para clasificación de páginas web

Arkaitz Zubiaga Mendialdua

Ingeniero en Informática

Mondragon Unibertsitatea

Director del Trabajo: Dr. Víctor Fresno Fernández

Departamento de Lenguajes y Sistemas Informáticos

ETS. de Ingeniería Informática - UNED

Índice general

Índice de figuras

Índice de tablas

Resumen

En este trabajo se presenta un estudio realizado sobre clasificación automática de páginas web, para lo que se han estudiado diferentes técnicas de aprendizaje automático, desde métodos supervisados hasta semisupervisados. Se ha considerado una tarea típica de clasificación de páginas web como un problema multiclase, donde el número de clases es mayor que dos, y como un problema de aprendizaje semisupervisado, ya que el porcentaje de documentos previamente etiquetados acostumbra a ser reducido en este entorno. Se escogieron las máquinas de vectores de soporte (Support Vector Machines, SVM), las cuales han demostrado en los últimos años una gran efectividad para clasificación automática de textos. No obstante, la naturaleza dicotómica y supervisada de esta técnica planteó la necesidad de trasladarla a este entorno semisupervisado, integrando la capacidad de entrenar con documentos no etiquetados, y teniendo en cuenta que la taxonomía definida es, a priori, mayor que la que presentan los problemas binarios. Así, este trabajo propone y compara diferentes aproximaciones, combinando tanto clasificadores semisupervisados binarios como supervisados multiclase, para resolver el problema planteado, mostrando los resultados obtenidos tras las experimentaciones realizadas sobre tres colecciones de páginas web de referencia. Las experimentaciones muestran que las combinaciones de clasificadores supervisados multiclase obtienen unos resultados notablemente superiores a las de semisupervisados binarios.

Capítulo 1

Introducción y objetivos

El aumento exponencial de la información disponible en formato digital durante los últimos años y las expectativas de crecimiento futuro hacen necesaria la organización de todo este contenido, con el fin de mejorar la búsqueda y acceso a la información. Con este fin, adquiere importancia la investigación de la clasificación automática de textos, y más especificamente la de páginas web.

La clasificación automática de textos se basaba, en sus inicios, en técnicas de ingeniería del conocimiento, donde un experto definía de forma manual las reglas que cada documento debía cumplir para pertenecer a una u otra categoría. No obstante, el gran coste que suponía esto, junto con los avances que se habían realizado en el área de la inteligencia artificial, dieron lugar en la década de los 80 a la utilización de técnicas de aprendizaje automático para estos propósitos Sebastiani (2002). Desde entonces, han sido muchos los métodos utilizados para clasificación automática de textos, una parte de los cuales se resumen en este trabajo.

El aprendizaje automático trata de obtener, de forma automatizada, las características que debe cumplir un objeto para ser clasificado en una determinada categoría, basándose para ello en una colección inicial de documentos preclasificados. Así, una vez que se han obtenido los descriptores para cada clase, el sistema los utiliza para crear un clasificador, pudiendo clasificar nuevos documentos.

Actualmente la mayoría de las técnicas para la construcción de clasificadores automáticos se basan en aprendizaje automático. Este aprendizaje puede ser de tres tipos diferentes, según su base de conocimiento:

- **Aprendizaje supervisado:** a partir de una colección de entrenamiento se aprenden las características que debe cumplir un documento para pertenecer a una u otra clase, creando posteriormente el clasificador. Una vez terminada esta fase de entrenamiento, el clasificador final está definido, el cual se utiliza para la categorización de documentos de los que no se conoce su clase.

- **Aprendizaje semisupervisado:** la fase de creación del clasificador utiliza la colección de entrenamiento como base, pero se sigue refinando con documentos sin clasificar. En estos casos, el número de documentos sin clasificar suele ser mucho mayor que el de los ya clasificados. Este tipo de aprendizaje hace que la disposición de un número reducido de documentos preclasificados no sea un problema para la clasificación automática, y se evite el trabajo costoso de tener que etiquetar o conseguir una gran colección preetiquetada, pero generalmente es más crítica la creación de un buen clasificador.

- **Aprendizaje no supervisado:** en estos casos se extraen los patrones de clasificación sin la disposición de una colección de documentos preclasificados. La clasificación es, por tanto, una agrupación en grupos sin etiquetar, lo que se denomina *clustering*. En el presente trabajo no se estudia este caso, al no considerarlo como un problema de clasificación propiamente dicho.

Asimismo, en los últimos años ha crecido notablemente el interés por aplicar estas técnicas de clasificación a las páginas web, lo que supone el traslado desde colecciones controladas de texto a un entorno tan heterogéneo y abierto como la Web. Ante esta problemática, se están aplicando las mismas técnicas que ya se venían utilizando para texto plano, aunque ahora surge la necesidad de considerar otros aspectos que antes no se tenían en cuenta.

Las técnicas de aprendizaje automático utilizadas inicialmente eran en su mayoría supervisadas, lo que supone la disposición de grandes colecciones de documentos previamente etiquetados para la fase de entrenamiento. La realidad es la que fuerza a la utilización de técnicas de aprendizaje semisupervisado, ya que las colecciones etiquetadas para páginas web suelen ser muy pequeñas en comparación con el tamaño de la Web.

En los últimos años, las máquinas de vectores de soporte (Support Vector Machines, SVM), en vista de los buenos resultados ofrecidos, se perfilan como una buena solución para los problemas de clasificación automática. Sin embargo, la naturaleza dicotómica y supervisada de este algoritmo de clasificación hace necesario e interesante el estudio de su aplicación a entornos semisupervisados multiclase como la Web.

Teniendo en cuenta todo ello, se ha considerado la tarea de clasificación de páginas web como un problema multiclase, donde generalmente se dispone de una taxonomía de más de dos categorías, y en la que la utilización de documentos no etiquetados para el aprendizaje mediante métodos semisupervisados podría resultar útil, ya que resulta complicado disponer de grandes colecciones previamente etiquetadas de páginas web.

Este trabajo presenta y compara diferentes aproximaciones a la utilización de SVM para clasificación de páginas web. Se centra en la obtención del algoritmo SVM semisupervisado multiclase apropiado, considerando ini-

cialmente el texto de las páginas, dejando para el futuro la inclusión de características propias de HTML.

El trabajo está estructurado de la siguiente manera. En el capítulo 2 (página 19) se expone el estado del arte en lo que a clasificación automática de textos se refiere, poniendo el foco en la tarea específica vinculada a las páginas web. Se presentan diferentes algoritmos que se han utilizado para clasificación automática, tanto en el ámbito del aprendizaje supervisado, como en el del aprendizaje semisupervisado. Además, se detallan las aplicaciones que originadas por este tipo de tareas, donde se muestra su utilidad.

En el capítulo 3 (página 53) se presenta el trabajo comparativo realizado durante este período. Se exponen las justificaciones que han llevado a la utilización de técnicas basadas en SVM, y se detallan las aproximaciones en las que se ha basado esta comparativa. Se describe toda la configuración, características y colecciones utilizadas para la experimentación llevada a cabo, y se presentan y analizan los resultados obtenidos en ésta.

Finalmente, en el capítulo 4 (página 75) se concluye con las ideas extraídas de la realización de este trabajo, explicando también las líneas que quedan abiertas para el futuro.

Capítulo 2

Clasificación automática de textos

En este capítulo se presenta una revisión de los trabajos que se han realizado hasta la actualidad en el área de la clasificación automática de textos, poniendo el foco especialmente en la clasificación de páginas web. Por un lado, se analizan los diferentes tipos de clasificación existentes y las características que pueden definir una tarea de este tipo, además de listar las múltiples aplicaciones que pueden hacer uso de estas técnicas. Por otro lado, se exponen las técnicas más conocidas que han sido utilizadas en la literatura, tanto para aprendizaje supervisado como para semisupervisado. Además, se presentan diversas técnicas utilizadas para la representación de los documentos y su posterior evaluación.

En general, se conoce como clasificación automática a la tarea de asignar una o varias categorías predefinidas sobre una colección de instancias a clasificar. Para ello, se asigna un valor booleano T o F para cada par (d_j, c_i), siendo d_j cualquier documento de la colección de documentos $D = d_1, ..., d_n$, y c_i cualquier categoría del conjunto de categorías predefinidas $C = c_1, ..., c_k$.

La tarea de la clasificación automática suele estar compuesta, generalmente, de las siguientes subtareas:

- **Representación:** Los documentos de texto o las instancias que componen la colección de datos a clasificar debe ser transformado a un formato comprensible para el sistema de clasificación que se vaya a utilizar. Para ello, se deben identificar las características representativas para los documentos, con el fin de que la representación sea adecuada (Ver sección 2.4).

- **Clasificación:** La subtarea de clasificación puede dividirse, a su vez, en dos fases: entrenamiento y test. En la fase de entrenamiento se alimenta el sistema de clasificación con los documentos que ya han pasado la fase de representación, con el fin de extraer los descriptores

de clase. Una vez realizado esto, la fase de test se ocupa de predecir las categorías correspondientes para los documentos por clasificar (Ver sección 2.5).

- **Evaluación:** Finalmente, tras haber predicho las categorías correspondientes a los documentos por clasificar, se procede a evaluar los resultados, para comprobar su calidad (Ver sección 2.6).

2.1. Clasificación textual vs clasificación de páginas web

La mayor parte de los estudios sobre clasificación documental se han centrado, hasta ahora, en texto plano, basándose sobre todo en las ocurrencias de los términos. No obstante, los estudios en torno a la clasificación de documentos estructurados, como es el caso de las páginas web, son mucho más reducidos, y no han tenido la debida atención hasta el momento.

En los trabajos realizados hasta la actualidad sobre clasificación de páginas web se han destacado múltiples características de este tipo de documentos que pueden resultar muy útiles:

- **Estructura del documento:** Kwon & Lee (2000, 2003) fueron unos de los primeros en dar importancia a las etiquetas HTML que componían las páginas web, mediante un método que las agrupaba en tres grupos, asignando diferentes pesos a cada uno de ellos. Otros trabajos posteriores han ido más allá. Kan (2004) y Kan & Thi (2005) demostraron que en algunos casos es posible clasificar documentos web disponiendo únicamente de su URL; esto puede resultar útil cuando puede resultar costoso obtener el contenido de todas las páginas. Golub & Ardō (2005), por su parte, utilizan cuatro partes de los documentos para el análisis: título, cabeceras, metadatos y el propio contenido. Los estudios realizados mostraron que una combinación de los cuatro elementos ofrecía los mejores resultados.

- **Estructura visual:** Además de la estructura que componen las etiquetas HTML, algunos estudios han ido más allá mediante el análisis de su visualización. Kovacevic *et al.* (2004) utilizan un grafo de adyacencia para representar la posición espacial de cada objeto HTML, ofreciendo mejores resultados que para el basado únicamente en ocurrencias de términos. Ven necesario el análisis de la estructura visual del documento, ya que diferentes etiquetados de HTML pueden dar lugar al mismo resultado visual.

- **Vecindad:** Ésta ha sido una de las características más utilizadas en el análisis de páginas web a la hora de su clasificación. En la mayoría

de los casos (Fürnkranz (1999, 2001); Glover *et al.* (2002); Sun *et al.* (2002); Cohen (2002)) se ha usado el texto con el que se enlaza desde otras páginas como información adicional, aunque también se ha utilizado en ciertas ocasiones (Attardi *et al.* (1999); Sun *et al.* (2002)) el texto íntegro que contenían aquellas páginas.

Otros estudios más recientes se basan en el contenido de más páginas, además de las que las enlazan directamente (padres). Calado *et al.* (2003) tienen en cuenta la clasificación de diversas páginas que la rodean para categorizarla: abuelos (los que enlazan a sus padres), padres, hermanos (aquéllos a los que sus padres también enlazan), hijos, nietos y cónyuges (aquéllos que también enlazan a los hijos) (ver figura 2.1). Angelova & Weikum (2006) y Qi & Davison (2006) se basan tanto en la clasificación correspondiente como en el contenido de las páginas que la rodean.

Varios estudios (Chakrabarti *et al.* (1998); Ghani *et al.* (2001); Yang *et al.* (2002)) muestran que la inclusión íntegra del contenido de los padres e hijos es perjudicial, y no beneficia en la clasificación, aunque esto no significa que aquellas páginas no puedan resultar útiles. Así, Oh *et al.* (2000) filtran aquellos padres e hijos que mantengan cierta similitud con la página objetivo.

Otra de las ideas que se ha estudiado últimamente es la clasificación de sitios web. Ester *et al.* (2002) presentaron un estudio en el que se mostraba que los títulos de las páginas que componen un sitio pueden ayudar de forma considerable en la clasificación de éste. No se ha realizado ningún estudio hasta el momento de forma inversa, donde se pruebe si la clasificación de un sitio web puede ayudar a la categorización de cada una de sus páginas.

Siguiendo con la clasificación automática de sitios web, Amitay *et al.* (2003) mostraron que existe una relación entre la estructura de enlaces que compone un sitio web y su funcionalidad. Basándose en esta misma idea, Lindemann & Littig (2006) analizaron esta relación y su aportación a la clasificación automática. Un análisis reciente sobre clasificación de páginas web lo realizan Qi & Davison (2007).

2.2. Tipos de clasificación automática

Dependiendo de los aspectos que se tengan en cuenta, la clasificación se puede catalogar en diferentes tipos, que se mencionan a continuación. Estos tipos son aplicables a la clasificación basada en el aprendizaje automático de forma general, y no únicamente a la clasificación de textos.

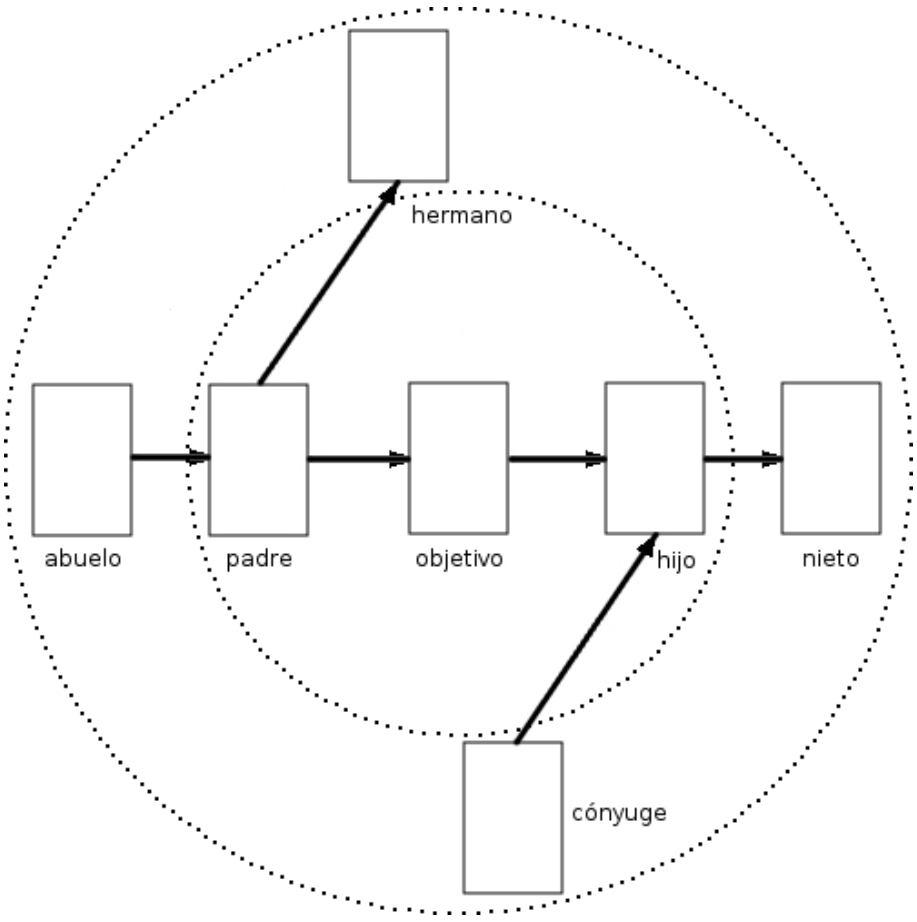

Figura 2.1: Muestra de vecindad para una página web

2.2.1. Clasificación single-label vs multi-label

Dependiendo de la aplicación, la clasificación de un documento puede variar en función de las categorías a las que pertenezca. Así, se denomina clasificación single-label cuando cada uno de los documentos de la colección debe tener una y sólo una categoría asignada, y clasificación multi-label cuando el número de categorías puede variar desde 0 hasta el número total de clases. Aplicaciones concretas de la clasificación multi-label pueden requerir que cada uno de los documentos corresponda a un número constante k de categorías.

Un caso concreto de la clasificación single-label es la clasificación binaria, asignando un T o F a cada documento. Así, una aplicación de filtro de spam asignará T a un documento si es spam, y F en caso de que no lo sea. De todas formas, la clasificación binaria se puede utilizar también en el caso del multi-label, asignando un T o F para cada par documento-categoría.

2.2.2. Clasificación sobre categorías vs clasificación sobre documentos

La clasificación se puede realizar analizando cada uno de los documentos y asignando sus correspondiente categorías (document-pivoted classification) o recorriendo cada una de las categorías e introduciendo todos los documentos relevantes en ella (category-pivoted classification). Es importante seleccionar el método de clasificación, dependiendo de si inicialmente se dispone de todos los documentos o no.

2.2.3. Clasificación 'hard' vs clasificación en ranking

A la hora de establecer la pertenencia de un documento a cierta categoría, existen dos posibilidades. Por una parte, se puede tomar una decisión 'hard', es decir, decidir directamente si un documento pertenece o no a una categoría; y por otra parte, se puede generar un ranking, en el que se posicionan los documentos con una puntuación según su adecuación a la categoría. En el primer caso se obtiene el listado no ordenado de documentos que pertenece a cada categoría, mientras que en el segundo el resultado es un ranking según los grados de coincidencia.

2.2.4. Clasificación fast-feature vs clasificación full-feature

Kules *et al.* (2006) distinguen dos tipos diferentes de clasificación dependiendo de las características en las que se basa. Se conoce como fast-feature cuando la única información utilizada para la clasificación es la que se incluye en el propio documento, sin tener en cuenta información externa al documento como puede ser el texto de los enlaces salientes. En el caso de la clasificación full-feature, sin embargo, se requieren características adicionales, externas al documento.

En el caso específico de la clasificación de documentos de texto, puede ser muy interesante la inclusión de técnicas full-feature cuando el texto puede tener su propia estructura y estar vinculado con otros documentos, como puede ocurrir en el caso del hipertexto en las páginas web, o los artículos de investigación que hacen referencia a otros. Fisher & Everson (2003) analizan cuándo es interesante la consideración de los enlaces entre documentos, mientras que Calado *et al.* (2003) proponen un método para combinar características textuales con los proporcionados por los enlaces.

2.2.5. Categorías simples (lean) y complejas (rich)

Kules *et al.* (2006) distinguen dos tipos de categorías. Denominan como simples (lean) a las categorías fácilmente clasificables, entre los que podrían estar, por ejemplo, en el caso de páginas web, categorías como el tipo de archivo, dominio al que pertenece, o diferentes rangos de tamaño de archivo.

Las categorías complejas (rich) agrupan a aquellas que clasifican taxonomías, ontologías o estructuras de conocimiento más complejas, para los que es necesario un mayor análisis del texto.

2.2.6. Clasificadores multilingües

La gran mayoría de los estudios relativos a la clasificación automática se han basado en colecciones de documentos en inglés, y todavía son pocos los estudios dirigidos a otras lenguas. Además, puede darse el caso de que las colecciones se compongan de textos en múltiples lenguajes, cuyo objetivo puede ser la de clasificarlos todos en la misma serie de categorías. Al hablar de colecciones de documentos multilingües, es probable que en muchos casos se desconozca el idioma en el que se encuentra cada texto, por lo que puede ser necesaria una fase previa de identificación de lenguaje.

Ante las diferentes posibilidades de clasificar los documentos de una colección multilingüe, García Adeva *et al.* (2005) comparan tres formas diferentes de hacerlo: un único procesamiento neutral de los documentos con un solo clasificador (utilizando una representación única en lengua ancla), procesamiento específico para cada lenguaje con un solo clasificador (representación independiente para cada lengua), y procesamiento específico para cada lenguaje con un clasificador para cada idioma. Los mejores resultados obtenidos fueron para este último caso. Otros estudios recientes en esta línea son de˜Melo & Siersdorfer (2007), Gliozzo & Strapparava (n.d., 2006), Rigutini *et al.* (2005), Amine & Mimoun (2007), Lee *et al.* (2006) y Lee & Yang (2007).

2.3. Aplicaciones de la clasificación automática

La clasificación automática se ha empleado en múltiples aplicaciones de diferentes tipos. A continuación se explican algunas de ellas.

2.3.1. Indexación automática para sistemas de recuperación de información

La indexación automática se suele utilizar basándose en un diccionario controlado, con lo que se extrae una serie de términos que podrían clasificar a cada uno de los documentos como si fueran palabras clave que los definen, mejorando así su acceso desde sistemas de recuperación de información. Esta aplicación está vinculada a la generación automática de metadatos.

2.3.2. Organización de documentos

Cuando se dispone de una colección de documentos muy grande, la clasificación de éstos de forma manual se convierte en una tarea que puede

resultar imposible. Es por ello que éste es un caso habitual de la utilización de técnicas de clasificación automática, que facilitan esta tarea de organización.

2.3.3. Filtrado de textos

Belkin & Croft (1992) definen el filtrado de textos como la clasificación de una serie de documentos entrantes mediante la distribución asíncrona de un productor de información hacia el receptor. Es habitualmente una clasificación del tipo single-label, en el que se diferencian los documentos relevantes e irrelevantes, aunque en algunos casos puede resultar interesante realizar una clasificación adicional entre los documentos no descartados. En el caso del filtrado adaptativo (Liddy *et al.* (1994)), el aprendizaje se realiza sobre cada uno de los perfiles de usuario, ampliando así las posibilidades de personalización.

Una de las aplicaciones más habituales del filtrado de textos es la de los sistemas de detección de spam, como en el caso de Androutsopoulos *et al.* (2000) y Drucker *et al.* (1999). Puede encontrarse un estudio sobre la materia en Gomez *et al.* (2004).

2.3.4. Desambiguación de significados de palabras (WSD)

Estos sistemas clasifican las palabras polisémicas según su significado, pudiendo determinar así a cual de los significados se refiere en cada uno de los casos. Éste es un problema de clasificación single-label. Gale *et al.* (1993), Escudero *et al.* (2000), Carpuat *et al.* (2004) y de˜Loupy *et al.* (2000) son algunos de los que han basado su sistema de desambiguación de palabras en técnicas de clasificación automática.

2.3.5. Directorios web

En el caso de un buscador web, puede resultar muy útil la disposición de una categorización jerárquica de las páginas web indexadas, también conocido como directorio web, con el fin de poder restringir la colección de documentos sobre los que aplicar la consulta. El método más utilizado suele ser la clasificación sobre categorías, dando posibilidad a la creación de nuevas y a la eliminación de otras innecesarias, para lo que muchas veces se establecen un número mínimo y un número máximo de documentos para cada categoría.

A la hora de clasificar páginas web, existen dos características de la Web que pueden tenerse en cuenta:

1. Estructura de la Web basada en enlaces, por lo que el contenido del documento referenciado por los enlaces puede afectar en la clasificación. Yang *et al.* (2002) ofrecen una comparativa de diferentes sistemas que

aprovechan esta característica, aunque son muy pocos los que lo hacen. Posteriormente, Calado *et al.* (2003), Lu & Getoor (2003), Fisher & Everson (2003) y Karamon *et al.* (2007), entre otros, han estudiado también esta característica.

2. Estructura jerárquica de las categorías, por lo que el problema se puede dividir en otros de menor dimensión. Dumais & Chen (2000), Ruiz & Srinivasan (2002) y Weigend *et al.* (1999) son algunos ejemplos que utilizan esta característica. Ceci & Malerba (2007) estudian este aspecto en profundidad.

2.3.6. Mejoras en la calidad de los resultados

Un sistema de categorización jerárquica de páginas web puede servir, a su vez, para mejorar los resultados ofrecidos por un buscador web, gracias a los metadatos que puede ofrecer. Chekuri *et al.* (1996) proponen un sistema de clasificación automática de páginas web, donde el usuario puede especificar la(s) categoría(s) sobre la que desea obtener resultados a la hora de la consulta sobre el buscador web. Este método puede funcionar cuando el usuario tiene claro la categoría sobre la que quiere buscar, aunque existen casos en los que el usuario no lo tiene tan claro y la técnica anterior puede no resultar tan útil.

Dumais & Chen (2000) proponen la muestra de resultados al usuario en forma de árbol, estructurados en categorías. Esto evita que el usuario deba seleccionar la categoría deseada, pero la necesidad de estructuración de los resultados en el momento de la consulta hace que no pueda ser tan eficiente como se hubiera deseado. Käki (2005) propone un sistema similar, mostrando que en muchos casos los usuarios pueden preferir visualizar los resultados de esta manera, en lugar del habitual ranking de resultados.

Otras técnicas como Haveliwala (2002) y Nie *et al.* (2006) añaden técnicas de clasificación automática al algoritmo de ranking PageRank (Page *et al.* (1998)), haciendo que éste sea sensible a la categoría que pertenece.

2.3.7. Soporte a sistemas de respuesta a preguntas

Un sistema de respuesta a preguntas puede utilizar técnicas de clasificación automática para mejorar sus resultados. Yang & Chua (2004b,a) proponen una solución para responder preguntas que solicitan listas de elementos, como puede ser la lista de estados de los Estados Unidos. Para ello, se realizan múltiples consultas sobre un buscador web, clasificando de la siguiente manera los resultados: páginas de colección (que contienen listas de elementos), páginas temáticas, páginas relevantes y páginas irrelevantes. Posteriormente, se realiza un proceso de clustering, extrayendo de ahí los resultados.

Otros estudios anteriores (Harabagiu *et al.* (2000); Hermjakob (2001); Kwok *et al.* (2001); Zhang & Lee (2003)) ya habían propuestos ideas similares.

2.3.8. Buscadores verticales o temáticos

Cuando se quieren desarrollar buscadores web verticales (específicos sobre un tema), realizar un crawling de toda la web es una tarea ineficiente. Ante esto, Chakrabarti *et al.* (1998) proponen una técnica denominada focused crawling, donde sólo se rastrean documentos relevantes sobre el tema deseado. Para ello, se emplea un clasificador automático para determinar la relevancia de un documento web.

2.3.9. Identificación de lenguaje

La Web es una gran colección de documentos en la que éstos se encuentran en muchos lenguajes diferentes, y en algunos casos puede resultar interesante la identificación del idioma. Una aplicación de la clasificación automática de textos puede ser la que clasifica éstos en función de su lenguaje, que ha ofrecido sus mejores resultados hasta la actualidad mediante la extracción de n-gramas. Así, Cavnar & Trenkle (1994) presentaron un algoritmo que posteriormente dio lugar a la implementación de TextCat, un sistema de identificación de lenguajes que ofrece muy buenos resultados, y que aún en la actualidad se sigue utilizando.

2.3.10. Clasificación de noticias

El gran número de noticias que existe en la Web en la actualidad es cada vez mayor, lo que complica su indexación y clasificación. La clasificación de noticias podría considerarse como una aplicación concreta de la clasificación de documentos, en la que los documentos a clasificar son noticias. Ésta es una de las lineas sobre las que más estudios se han realizado hasta la actualidad, debido a la existencia de multitud de colecciones de documentos etiquetados. Algunos de los sistemas más conocidos de este tipo son Google News y Newsblaster. El primero, además de clasificar las noticias por temas, agrupa noticias de diferentes medios que tratan de lo mismo, facilitando la lectura al usuario.

Algunos ejemplos de estos tipos de sistemas se proponen en Mittermayer & Knolmayer (2006), Wermter & Hung (2002) y Maria & Silva (2001), mientras que Billsus & Pazzani (2000) y Dörfler *et al.* (2001) presentan soluciones adaptativas al perfil del usuario.

2.3.11. Recuperación de información translingüe (CLIR)

Recientemente han crecido de forma considerable los estudios en clasificación automática con documentos en diferentes lenguas como colección a clasificar (ver sección 2.2.6). Esta nueva línea de investigación ha dado lugar a la creación de nuevas aplicaciones basadas en la clasificación automática, que amplían sus posibilidades a entornos multilingües. Entre las aplicaciones destacables está la de Ruch (2004), que propone basar la traducción de consultas de un sistema de recuperación de información translingüe en la clasificación de éstas mediante términos listados de un vocabulario controlado. No obstante, éste ha sido un caso único hasta el momento, que no ha dado lugar a nuevos experimentos.

2.3.12. Clasificación de sentimientos

La clasificación automática se centra, habitualmente, en los temas que tratan los documentos. No obstante, cuando esta clasificación se realiza basándose en el enfoque de los documentos, se denomina clasificación de sentimientos. Algunos tipos de sentimientos por los que se puede clasificar pueden ser, por ejemplo, la positividad o negatividad de una opinión, o una actitud o situación emocional ante algo.

Entre los estudios realizados hasta la actualidad, Kennedy & Inkpen (2005, 2006) presentan un método para la clasificación sentimental de críticas sobre películas y otro tipo de productos, mientras que Cui *et al.* (2006) simplifican el problema presentando una solución para clasificar críticas positivas y negativas. Otros ejemplos se pueden encontrar en Aue & Gamon (2005), Goldberg & Zhu (2004), Pang & Lee (2005), Pang *et al.* (2002), Turney (2001), y Blitzer *et al.* (2007).

2.4. Representación

La representación del contenido de un documento es la fase en la que un texto escrito en lenguaje natural se transforma en un formato que posibilite su tratamiento y posterior clasificación. La indexación, por su parte, es la extracción de rasgos (habitualmente términos) para cada documento y su posterior almacenamiento en un formato fácilmente procesable por el sistema.

La problemática de la representación se centra en la captura del contenido de los documentos. La elección de una forma de representación para el texto incluye tanto la selección de las unidades textuales como la de las reglas de combinación de estas unidades. No obstante, esta última no se tiene en cuenta en muchos casos. Así, las principales diferencias entre los métodos de representación se suelen limitar a los siguientes aspectos:

- **Diferentes tratamientos de lo que es un término.** Lo más habitual es seleccionar como términos a palabras únicas, y en algunos experimentos (Apté *et al.* (1994); Dumais *et al.* (1998); Lewis (1992)) se ha demostrado que otras opciones no ofrecen unos resultados muy superiores. Una buena opción puede ser la combinación de diferentes métodos de indexación (Tzeras & Hartmann (1993)). Yu & Liu (2004) proponen un método basado en la selección de rasgos que caracterizan a los documentos según su relevancia y redundancia, y Mejía-Lavalle *et al.* (2006) proponen un nuevo método que añade eficiencia.

- **Diferentes formas de asignar pesos a los términos.** Los pesos asignados a cada uno de los términos suelen ser entre 0 y 1, para lo que el método más utilizado es el tfidf[1], con lo que cuanto más frecuente es un término en un documento, más relevante es, y cuanto más documentos contienen el término, menos discriminatorio es y, por tanto, menos relevante respecto al contenido textual. Uno de los problemas de este sistema es que no se tiene en cuenta la semántica de los términos, con lo que no tiene ninguna importancia la posición en la que salen ni la relación con otros términos.

Otras técnicas pueden ser utilizadas cuando no se dispone inicialmente de todos los documentos, y por lo tanto es imposible obtener el valor idf.

Antes de realizar estas operaciones, se utilizan técnicas de eliminación de stopwords, que son aquellas palabras muy comunes en el texto que no resultan discriminantes para este tipo de tareas, y en algunos casos de *stemming*, que es una aproximación a la extracción de la raíz de las palabras, aunque en algún caso se ha demostrado que el *stemming* puede afectar de forma negativa en el rendimiento del sistema (Baker & McCallum (1998)).

2.4.1. Darmstadt Indexing Approach

Darmstadt Indexing Approach (DIA) es el método de indexación que se utilizó para el proyecto AIR (Fuhr *et al.* (1991)). Este método se basa en la indexación de términos incluidos en un vocabulario controlado. Posteriormente, se tienen en cuenta ciertas propiedades (de términos, de la relación entre término y documento, de los documentos, y de las categorías). De esta manera, se crean unos vectores de descripción de relevancia para cada par documento-categoría.

Sin embargo, no se volvió a utilizar esta técnica con posterioridad, por lo que no existen referencias más recientes.

[1] El sistema de pesado tf-idf (term frequency-inverse document frequency) es una medida estadística que evalúa la importancia de un término en una colección.

2.4.2. Representación basada en n-gramas

Los n-gramas son cadenas de n caracteres que aparecen uno seguido de otro, que pueden ser parte de una cadena mayor. Cavnar & Trenkle (1994) propusieron un método basado en la similitud de n-gramas para la clasificación automática de textos. Se realizaron pruebas tanto con fines de identificación de idioma del texto como para clasificación por temas, ofreciendo unos buenos resultados. Este algoritmo puede resultar útil, sobre todo, en el caso de identificadores de idioma y sistemas de filtrado.

Nather (2005) presentó un nuevo sistema basado en n-gramas para la clasificación temática de textos, con peores resultados de los esperados, aunque la falta de disponibilidad de una buena colección de documentos hace que las pruebas no fueran suficientes.

2.4.3. Reducción de dimensión

El número de términos encontrado en una colección puede ser muy amplio, y por eso puede resultar interesante en algunos casos su reducción. Fuhr & Buckley (1991) proponen que el número de documentos en la colección por cada término debe estar entre 50 y 100. Esto hace que el peligro de sobreajuste (overfitting) sea menor, es decir, hace que el entrenamiento no converja demasiado rápido o que éste se estanque debido a un ajuste excesivo. Aún así, se debe de tener cuidado con no eliminar información relevante.

La reducción de dimensión puede ser de forma local o global:

- **Reducción local:** cuando se selecciona un conjunto reducido de términos para cada una de las categorías.
- **Reducción global:** cuando se selecciona un conjunto reducido de términos para el conjunto de todas las categorías.

Otra distinción se puede destacar en la forma de realizar la reducción:

- Reducción de dimensión por reducción de términos.
- Reducción de dimensión por extracción de términos.

2.4.3.1. Reducción de dimensión por reducción de términos

Se trata de reducir el número de términos que componen la colección mediante la selección de los términos que ofrecen una mejor efectividad. Dos son las aproximaciones más conocidas para ello:

- **Aproximación superficial** (wrapper approach)(Moulinier *et al.* (1996), Kohavi & John (1998)). Cada vez que se extrae o se añade un término

del conjunto, el mismo método de aprendizaje que servirá para la clasificación es utilizado para su evaluación, por lo que se estima si el nuevo conjunto de términos es mejor que el anterior o no. La colección de términos que mejores resultados da es la escogida. Tiene la ventaja de que la evaluación de la colección de términos se basa en el mismo método de aprendizaje que servirá para el sistema de aprendizaje, pero el gran número de colecciones de términos que se deben evaluar hacen que sea muy costoso computacionalmente. La idea de esta técnica se ha seguido utilizando hasta la actualidad, como por ejemplo en Farmer & Jain (2004).

- **Aproximación por filtrado** (filtering approach). Algunos términos se extraen mediante la aplicación de una función que hace de filtro que mide la importancia de los términos, que puede ser:

  - Frecuencia documental: sólo se mantienen los términos que ocurren en un gran número de documentos. Yang & Pedersen (1997) demostraron que para la colección de documentos que emplearon se puede reducir la dimensión hasta una décima parte, sin perder efectividad.
  - Funciones probabilísticas: también se pueden utilizar métodos probabilísticos, que se basan en que los términos óptimos para cada categoría son aquéllos que están distribuidos de forma diferente en los ejemplos positivos y en los negativos.

  Varios experimentos (Caropreso *et al.* (2001); Galavotti *et al.* (2000); Mladenić (1998); Yang & Pedersen (1997)) han comparado ambos métodos de reducción por filtrado, y son muchas las funciones probabilísticas que han superado los resultados de la frecuencia documental.

2.4.3.2. Reducción de dimensión por extracción de términos

Teniendo en cuenta los problemas que pueden originar la polisemia, la homonimia y la sinonimia, mediante la extracción de términos se crean nuevos términos que pueden no existir en el documento, pero que no varían su semántica. Existen diferentes métodos para ello:

- **Clustering de términos:** el objetivo del clustering de términos es la agrupación de palabras con grandes parecidos semánticos. Tanto Lewis (1992) como Li & Jain (1998) han utilizado técnicas de clustering, las cuales no se ha demostrado que tengan un resultado superior que sin su utilización. Baker & McCallum (1998) definieron técnicas de clustering que utilizan información previa, realizando agrupaciones de términos para aquellos que coinciden en categorías; este último, junto con posteriores pruebas de Slonim & Tishby (2001), demostró tener

únicamente un 2 % de reducción en efectividad. Algunos estudios recientes (Wang & Wang (2005)) incluyen también esta técnica.

- **Latent Semantic Indexing:** esta técnica se basa en la representación de los documentos teniendo en cuenta la coocurrencia de términos como factores que permiten reducir el número de dimensiones. Lo que se obtiene como resultado es una representación vectorial de una dimensión inferior, gracias a las agrupaciones de términos que coaparecen en los documentos. Wiener *et al.* (1995) y Schütze *et al.* (1995) han presentado diferentes soluciones, que han demostrado obtener resultados muy superiores a los de las técnicas de reducción de términos; además, la extracción de términos local (realizándose sobre cada una de las categorías) dio mejores resultados que la extracción de términos global (sobre todas las categorías de forma conjunta). Éste es, actualmente, uno de los métodos de indexación más utilizados, debido a sus buenos resultados, con ejemplos como Lee & Yang (2007), Wang *et al.* (2005) o Slonim & Tishby (2001).

- **Utilización de ontologías:** Algunos estudios recientes proponen el uso de ontologías para la indexación de los documentos y poder reducir así su dimensión terminológica mediante la búsqueda de relaciones entre términos. Aunque técnicas como el Latent Semantic Indexing ofrecen buenos resultados, Wu *et al.* (2003) aseguran que los resultados de un sistema basado en ontologías pueden ser superiores, siendo posible, además, la edición manual de los índices, lo que puede facilitar la realización de pequeñas correcciones; de˜Melo & Siersdorfer (2007) utilizan también técnicas basadas en ontologías.

2.5. Algoritmos de clasificación

2.5.1. Colección inicial

Las técnicas de aprendizaje automático centran su base de aprendizaje en el corpus inicial de documentos que se dispone. Una vez que el clasificador ha sido construido, es conveniente poder validarlo, por lo que antes de generar el clasificador se divide la colección inicial en dos subcolecciones:

- Subcolección de entrenamiento: estos documentos son utilizados para analizar sus características y poder crear así el clasificador.

- Subcolección de prueba o test: se utiliza para comprobar la calidad de los clasificadores generados tras la fase de entrenamiento.

Además de la comprobación utilizando la colección de prueba, se suele realizar un test adicional con la colección inicial completa. Los resultados

de todas estas fases deben ser similares, lo que confirma su correcto funcionamiento ante diferentes subcolecciones. Esta técnica se conoce como entrenamiento-y-prueba (train-and-test).

Basándose en la anterior técnica, se utiliza frecuentemente también la denominada k-fold cross-validation, que construye k clasificadores diferentes, para después aplicar el entrenamiento-y-prueba a cada uno de ellos, quedándose con el que mejores resultados ofrece o realizar una combinación entre ellos.

En algunos casos se puede llegar a dividir la colección de entrenamiento en dos colecciones más pequeñas, creando una nueva colección de validación. En estos casos, se utiliza la colección de entrenamiento para la generación del clasificador, y la colección de validación para su posterior optimización; finalmente, se comprueban los resultados con la colección de prueba.

También se puede obtener el valor de generalidad (generality) para cada una de las categorías, siendo éste el valor que define el porcentaje de documentos de la colección que pertenece a tal categoría. Se pueden obtener valores parejos para cada una de las subcolecciones definidas.

Para la definición del umbral que establece las categorías que corresponden o no a cada documento, son tres las técnicas más utilizadas, como explica Yang (2001):

- **RCut**, o **umbralización basada en el ranking**: para cada uno de los documentos, se ordenan las categorías según la puntuación obtenida, para después quedarse con las k mejor posicionadas como candidatas.
- **PCut**, o **umbralización proporcional**: para cada una de las categorías, se ordenan los documentos por puntuación, multiplicándolo después por un valor k_j. El valor de k_j se define de la siguiente manera:

  $$k_j = P(c_j) \cdot x \cdot m$$

  donde:

  $P(c_j)$ es la probabilidad que tiene cualquier documento de pertenecer a la categoría j,

  x es el número medio de categorías que el sistema establecerá a cada documento, y

  m es el factor que representa el tamaño de la colección de documentos

- **SCut**, o **umbralización basada en la puntuación**: cuando se realiza la fase de validación con la colección de validación, se puntúan los documentos para cada una de las categorías, y se definen unos umbrales locales en base a esas puntuaciones.

2.5.2. Técnicas de aprendizaje supervisado

La gran mayoría de los estudios realizados hasta la actualidad en torno a la clasificación automática se ha basado en técnicas de aprendizaje supervisado, para lo cual se han utilizado múltiples algoritmos diferentes. Estos métodos son muy eficientes cuando se dispone de una colección considerable de instancias etiquetadas, aunque pueden haber problemas cuando el tamaño de ésta es pequeña sobre la colección completa a etiquetar.

2.5.2.1. Clasificadores probabilísticos

Los clasificadores probabilísticos se basan en el teorema de Bayes para establecer si un documento pertenece a una categoría:

$$P(c_i|\vec{d_j}) = \frac{P(c_i) \cdot P(\vec{d_j}|c_i)}{P(\vec{d_j})}$$

Sin embargo, la dificultad de calcular el valor $P(c_i|d_j)$ hace que en muchos casos se tenga que recurrir a la *suposición de independencia* que asume el modelo espacio vectorial (VSM), lo que conlleva que dos coordenadas cualquiera del vector de representación del documento son estadísticamente independientes. Aunque en la teoría la aplicación de la suposición de la independencia no es del todo correcta, en la práctica facilita en gran medida la complejidad del clasificador, y la reducción del rendimiento es muy pequeña. A este algoritmo se le denomina el clasificador de Naïve Bayes, utilizado, entre otros, por Joachims (1998), Li & Jain (1998) y Lewis (1992):

$$P(d_j|c_i) = \prod_{k=1}^{|T|} P(\omega_{kj}|c_i)$$

2.5.2.2. Clasificadores basados en árboles de decisión

En el contexto de la clasificación automática de textos, un árbol de decisión (Mitchell (1997), capítulo 3) es un árbol cuyos nodos internos están etiquetados por términos, las ramas salientes están etiquetadas por los pesos de éstas, y las hojas corresponden a categorías. Así, se recorre el árbol de arriba a abajo para cada uno de los documentos, hasta llegar a una de las hojas, es decir, hasta asignar una categoría.

Fuhr *et al.* (1991), Lewis & Catlett (1994), Lewis & Ringuette (1994), Cohen & Singer (1999), Joachims (1998), Li & Jain (1998), Schapire & Singer (2000), y Weiss *et al.* (1999) son algunos ejemplos de utilización de clasificadores basados en árboles de decisión. Ésta es, no obstante, una técnica que no se utiliza apenas en la actualidad.

2.5.2.3. Clasificadores basados en reglas

Los clasificadores basados en reglas son muy similares a los basados en árboles de decisión, con la diferencia de que se trata de definir una serie de reglas DNF (forma normal disyuntiva) para cada una de las categorías. Así, se consigue una forma más compacta de clasificar los documentos, sin la necesidad de recorrer el árbol completo.

Moulinier & Ganascia (1996), Li & Yamanishi (2002), Cohen (1995), Cohen & Hirsh (1998), Cohen & Singer (1999), Moulinier *et al.* (1996) y Apté *et al.* (1994) han presentado diferentes soluciones con clasificadores basados en reglas. Posteriormente, la utilización de este método se redujo de forma considerable, dando lugar al uso de otras técnicas con menor necesidad de interacción por parte de expertos.

2.5.2.4. Métodos de regresión

La función ideal de clasificación es aquélla que clasificaría todas las instancias de forma correcta. Así, los métodos de regresión se basan en la creación de una aproximación a la función ideal de clasificación con valores reales en lugar de binarios, mediante la que se ajustan los valores de entrenamiento. El método de regresión más conocido es el *Linear Least-Squares Fit*, empleado por Yang & Chute (1994). En él, se asignan dos vectores a cada documento: un vector de entrada que representa los pesos de los términos, y otro vector de salida que representa los pesos de las categorías. Con estos vectores se crean unas matrices para que posteriormente sean procesadas. La clasificación se realiza mediante la multiplicación de ambas matrices (categorías y términos). Aunque se ha demostrado que este método ofrece muy buenos resultados, los costes de computación son muy elevados; es por ello que hoy en día apenas se utiliza esta técnica.

2.5.2.5. Métodos on-line

Un clasificador lineal utiliza un vector para cada categoría existente y otro para cada documento, los cuales están formados por los pesos de cada uno de los términos de la colección sobre esa categoría o documento: $\vec{c_i} = \langle \omega_{1i}, ..., \omega_{|T|i} \rangle$ y $\vec{d_j} = \langle \omega'_{1j}, ..., \omega'_{|T|j} \rangle$, siendo $|T|$ el número de términos existente. Así, el valor que un documento dj obtiene para cierta categoría ci es el resultado del sumatorio $\sum_{k=1}^{|T|} \omega_{ki} \cdot \omega'_{kj}$, teniendo en cuenta el peso que cada uno de los términos tiene sobre ese documento y sobre esa categoría. Este cálculo debe realizarse para cada par documento-categoría. Entre los métodos lineales se pueden distinguir los que se ejecutan por lote (batch), donde se analiza toda la colección de entrenamiento a la vez, y los métodos on-line (o incremental), donde se empieza a construir el clasificador con el primer documento de entrenamiento, y se va refinando conforme se van analizando nuevos. Uno de los métodos más comunes entre los incrementales es

el *perceptron* (Dagan *et al.* (1997); Ng *et al.* (1997)), que va ajustando los pesos según va descubriendo nuevas entradas, sumando para casos positivos, y restando para casos negativos.

2.5.2.6. Métodos Rocchio

El método Rocchio se basa en un clasificador lineal que utiliza perfiles. Este método lo introdujo Hull (1994) en la clasificación automática de textos, que fue utilizado con frecuencia posteriormente (Joachims (1997, 1998); Sable & Hatzivassiloglou (2000); Cohen & Singer (1999); Larkey & Croft (1996)), aunque actualmente ya no se utiliza mucho. Esta técnica se basa en el acercamiento hacia los casos positivos ($|POS|$), y el alejamiento de los casos negativos ($|NEG|$), asignando una ponderación para cada uno de estos casos; un valor β asigna un peso a la aproximación hacia los positivos, mientras que un valor γ lo hace para alejarse de los negativos. Para ello, se realiza una suma de los casos positivos y una resta de los casos negativos, ambos con su correspondiente ponderación, definido por la siguiente fórmula:

$$\omega_{ki} = \beta \cdot \sum_{d_j \in |POS_i|} \frac{\omega_{kj}}{|POS_i|} - \gamma \cdot \sum_{d_j \in |NEG_i|} \frac{\omega_{kj}}{|NEG_i|}$$

Una de las mejoras que se puede realizar a la fórmula de Rocchio es la de tener en cuenta los ejemplos que están cerca de los positivos ($|NPOS|$), en lugar de todos los negativos, con lo que la fórmula resultaría de la siguiente manera:

$$\omega_{ki} = \beta \cdot \sum_{d_j \in |POS_i|} \frac{\omega_{kj}}{|POS_i|} - \gamma \cdot \sum_{d_j \in |NPOS_i|} \frac{\omega_{kj}}{|NPOS_i|}$$

El método de Rocchio se ha utilizado para el denominado *Relevance Feedback*, es decir, el sistema que utilizan los motores de búsqueda web para mejorar los resultados de búsqueda según las opciones seleccionadas por los usuarios (Buckley *et al.* (1994)).

2.5.2.7. Redes neuronales

Una red neuronal es una red de unidades, donde las unidades de entrada representan diferentes características, las de salida representan las categorías resultantes, y el conjunto de pesos de las conexiones representa las relaciones. Un método conocido es el *perceptron* (ver sección 2.5.2.5).

Un método de entrenamiento de redes neuronales es la *retropropagación*, donde se cargan los pesos de los términos como unidades de entrada, y se retrocede en caso de una clasificación incorrecta, volviendo a probar cambiando los parámetros de la red, con el fin de minimizar el error.

Por otra parte, las *redes neuronales no lineales* (Lam & Lee (1999), Ruiz & Srinivasan (2002), Weigend *et al.* (1999), Yang & Liu (1999), Ruiz & Srinivasan (2002)) son aquellas que disponen de una o más capas entre las unidades de entrada y de salida. Sin embargo, esto parece que no ofrece mejoras (Schütze *et al.* (1995)), o éstas son mínimas (Wiener *et al.* (1995)), respecto a las redes lineales.

Otros trabajos que utilizan redes neuronales para el proceso de aprendizaje son Selamat & Omatu (2004) y Chen *et al.* (2005).

2.5.2.8. Clasificadores basados en ejemplos

Los métodos basados en ejemplos clasifican los documentos en sus categorías correspondientes según sus parecidos a los documentos de entrenamiento utilizados como ejemplos. En este aspecto, el método más utilizado para la clasificación automática ha sido el denominado *k-Nearest Neighborhood*, o simplemente *k-NN*. Este algoritmo, a la hora de decidir qué documentos pertenecen a cierta categoría, seleccionan los k documentos más similares al utilizado como ejemplo. Por tanto, este método requiere la definición previa de un valor k.

Galavotti *et al.* (2000) propusieron una variante de *k-NN*, en el que no se descarta la información cuando un documento no pertenece a cierta categoría, sino que esta puntúa negativamente.

Por otro lado, Lam & Ho (1998) presentaron un método que combina el uso de ejemplos con los perfiles, basándose en instancias generalizadas (GIs) en lugar de entrenar documentos. En él, se realiza el siguiente proceso:

- Agrupar mediante clustering la colección de entrenamiento.
- Crear un perfil G(k_{iz}) para cada grupo, mediante algún algoritmo de aprendizaje lineal.
- Aplicar k-NN sobre los perfiles en lugar de entrenar los documentos.

2.5.2.9. Máquinas de Vectores de Soporte o Support Vector Machines (SVM)

Joachims (1998, 1999) utilizó las máquinas de vectores de soporte por primera vez en clasificación automática. Las máquinas de vectores de soporte se basan en un espacio de dimensión $|T|$, siendo $|T|$ el número de términos existente, y tratan de obtener la superficie o capa que distingue los ejemplos positivos de los negativos para cada categoría en ese espacio de dimensión $|T|$. Para establecer estas superficies de decisión se utiliza una colección de ejemplos de entrenamiento, denominados vectores de soporte; así, de todas las posibilidades obtenidas como resultados, se escoge la capa que mayor margen tiene sobre las clases.

Para el caso de un problema con únicamente dos categorías separables linealmente, la función que decida la clasificación de cada instancia será del siguiente tipo:

$$f(x) = \omega \cdot x + b$$

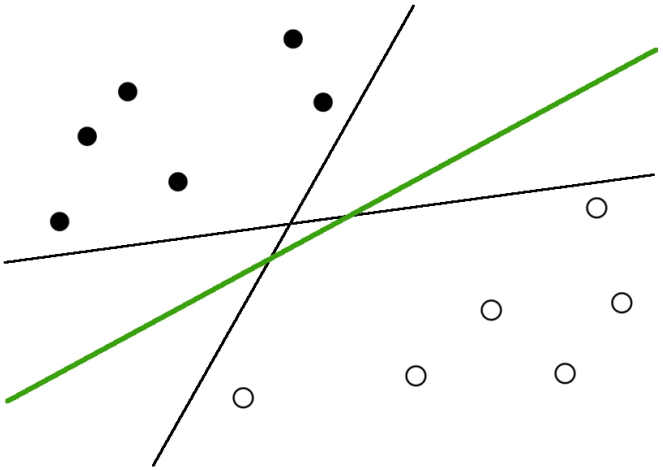

Figura 2.2: Clasificación con SVM

En el ejemplo de la figura 2.2, se muestra la representación y selección de dicha línea, donde el resultado es aquél que maximiza el margen hacia los puntos más cercanos de cada una de las dos clases existentes. Además, se tiene en cuenta que ningún punto se encuentre dentro de este margen.

La función anteriormente citada es muy difícil de optimizar computacionalmente, por lo que el problema se suele centrar en la optimización de la siguiente función equivalente (Boser *et al.* (1992), Cortes & Vapnik (1995)):

$$\text{mín}\, \frac{1}{2}||\omega||^2 + C \cdot \sum_{i=1}^{n} \xi_i^d$$

Sujeto a:

$$y_i(\omega \cdot x_i + b) \geq 1 - \xi_i, \xi_i \geq 0$$

donde:

ξ es el margen de error entre el punto i y la capa de separación,

C es el parámetro definido para asignar un peso sobre cada categoría, y

d tiene un valor de 1 para un coste lineal, y un valor de 2 para un coste cuadrático

De manera similar al ejemplo anterior, cualquier problema linealmente separable que se aplique a un espacio de dimensión $|T|$ puede resolverse mediante una capa de dimensión $|T| - 1$.

Aunque el anterior ejemplo muestre un problema lineal, las máquinas de vectores de soporte pueden utilizarse también cuando las categorías no son linealmente separables. Es necesaria la utilización de una función de kernel para estos casos, que sirva para definir la separación.

Las máquinas de soporte vectorial ofrecen, principalmente, dos ventajas a la clasificación automática Joachims (1998):

- No se requiere la selección de términos, aunque su aplicación puede resultar interesante.
- No es necesario realizar un esfuerzo de ajuste de parámetros en el caso de problemas linealmente separables, ya que dispone de su propio método para ello.

Las máquinas de soporte vectorial son actualmente el método más utilizado para el aprendizaje automático en sistemas de clasificación automática de textos. Entre los estudios recientes realizados mediante la utilización de este método se encuentran Kolcz & Chowdhury (2007), Lee & Yang (2007), Qi & Davison (2006), y Mittermayer & Knolmayer (2006). Cabe destacar que las SVM se limitan, en principio, a la clasificación binaria, donde únicamente pueden existir dos clases, aunque tal y como se explicará en el capítulo 3, existen soluciones para problemas multiclase.

2.5.2.10. Clasificadores unificados (Classifier Commitees)

La idea de los clasificadores unificados es la de construir un sistema de clasificadores, combinando posteriormente los resultados obtenidos de todos ellos. Esto supone en primer lugar la selección de un número determinado k de clasificadores, y la decisión de un método para combinar los resultados. Se supone que cada uno de los clasificadores devuelve un valor CSV (Classification Score Value) para cada par documento-categoría, que valora su pertenencia o no.

La selección de los clasificadores puede ser totalmente libre en cada uno de los casos, escogiendo algunos que den buenos resultados. En cuanto a la decisión del método de combinación de resultados, son varias las opciones que se han utilizado hasta ahora:

- **Por mayoría absoluta:** se obtienen los valores binarios devueltos por los clasificadores, y la opción con más votos (0/1) será escogida.

- **Combinación lineal de los pesos:** la suma de los valores CSV de todos los clasificadores da como resultado el valor CSV global, lo que se utiliza para la clasificación final.
- **Selección dinámica del clasificador:** de todos los clasificadores utilizados, se comprueba cuál de ellos es el más eficiente en la fase de validación, y su decisión es la que se toma por válida.
- **Combinación adaptativa de clasificadores:** se suman las decisiones de todos los clasificadores, pero su valor es ponderado en función del resultado obtenido en la fase de validación por cada clasificador.

Una variante de los clasificadores unificados es el *boosting* (Schapire *et al.* (1998); Schapire & Singer (2000)), donde todos los clasificadores aprenden bajo el mismo método de aprendizaje, y la unión de éstas se va realizando de forma incremental, construyendo un clasificador más firme mediante la unión de varios sencillos.

2.5.2.11. Otras técnicas de clasificación supervisada

Aunque en esta sección se han comentado las técnicas más habituales para la clasificación automática de documentos, también se han utilizado otros métodos como las redes de inferencia bayesianas (Dumais *et al.* (1998), Lam *et al.* (1997), Tzeras & Hartmann (1993)), los algoritmos genéticos (Clack *et al.* (1997), Masand (1994)) o el modelado de entropía máxima (Manning & Schütze (1999)).

2.5.3. Técnicas de aprendizaje semisupervisado

En el aprendizaje semisupervisado se dispone, generalmente, de una colección limitada de instancias etiquetadas, por lo que un aprendizaje adicional al supervisado es requerido. Así, estas técnicas de aprendizaje utilizan la colección no etiquetada para seguir aprendiendo y, por lo tanto, refinando el clasificador. La disposición de grandes colecciones de documentos y la gran dificultad para el proceso de etiquetado de una parte significativa de éstos, hace que estas técnicas sean cada vez más utilizadas.

Entre las diferentes técnicas de aprendizaje semisupervisado se pueden distinguir las transductivas y las inductivas:

- **Aprendizaje inductivo:** (ver figura 2.3) tras la fase de aprendizaje realizada sobre la colección de entrenamiento, se genera un modelo, lo que se conoce como inducción. Posteriormente, este modelo se utiliza para clasificar la colección de test, a lo que se denomina deducción. El inconveniente de este tipo de técnicas de aprendizaje es que al definir un modelo fijo, éste no se va refinando en la fase de test, aunque por su

parte positiva ofrece la posibilidad de manejar documentos no vistos, ya que existe este modelo definido.

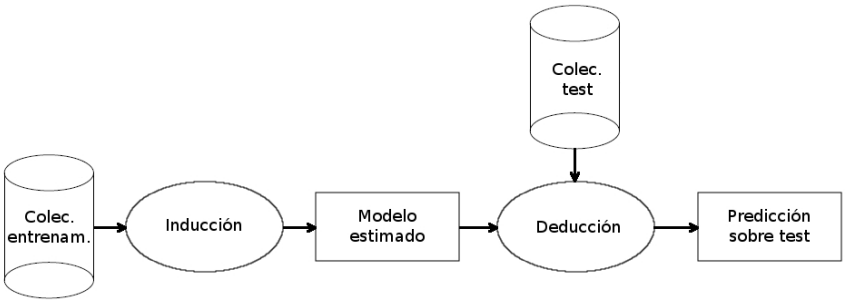

Figura 2.3: Aprendizaje inductivo

- **Aprendizaje transductivo:** (ver figura 2.4) en este tipo de aprendizaje, la función de clasificación se va modificando mientras se va procesando la colección de entrenamiento, y se sigue refinando cuando se procesa la colección de test. Este proceso se conoce como inferencia transductiva. Ofrece la ventaja del refinamiento continuo durante la fase de test, aunque este tipo de aprendizaje no es capaz de manejar posteriormente documentos no vistos, al no existir un modelo previamente definido.

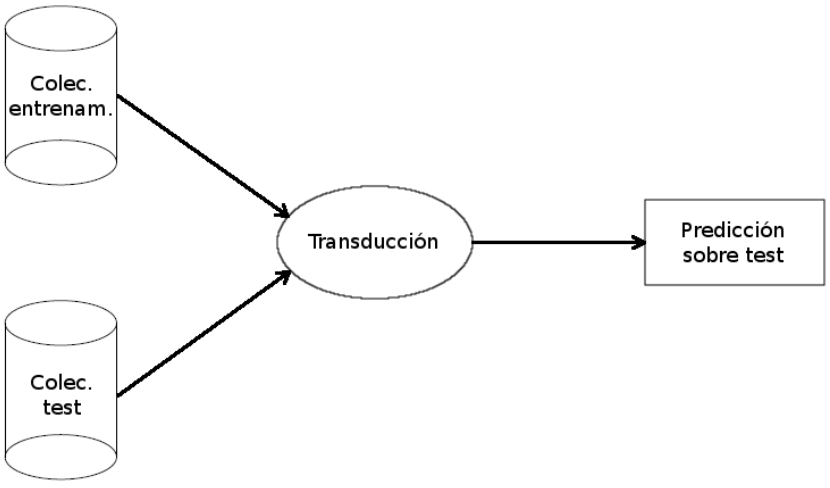

Figura 2.4: Aprendizaje transductivo

2.5.3.1. Autoentrenamiento

El método de autoentrenamiento es uno de los primeros que se comenzó a utilizar entre las técnicas semisupervisadas, también conocido con el nombre de *bootstrapping*. En primer lugar se realiza un entrenamiento básico basado en la pequeña colección de documentos etiquetados de la que se dispone. Posteriormente, se va incrementando eses entrenamiento mediante la selección de los resultados más *fiables* entre los documentos no etiquetados. Se repite este proceso hasta que todos los documentos están etiquetados.

Éste es el método más simple de los que se utiliza en el entorno semisupervisado, de uso frecuente en el área del procesamiento del lenguaje natural, aunque su mayor problema es que los errores iniciales realizados pueden dar lugar a la convergencia hacia una clasificación errónea.

Esta técnica ha sido utilizada por Yarowsky (1995) para desambiguación de significados, por Riloff *et al.* (2003) para identificación del sujeto y por Maeireizo *et al.* (2004) para clasificación de diálogos como emocionales o no emocionales, entre otros.

2.5.3.2. Co-entrenamiento

Para aplicar la técnica de co-entrenamiento (Blum & Mitchell (1998)), previamente se realiza una de división de las características, para lo que se supone que:

- Las características se pueden agrupar en dos subconjuntos.
- Cada uno de los subconjuntos de características puede ser válido para realizar la clasificación.
- Cada uno de los subconjuntos de características es independientes sobre el otro.

Un ejemplo de este tipo de características se puede dar en el caso de las páginas web, donde una de las características puede ser el propio contenido de la página, y la otra puede ser el texto con el que otras páginas la enlazan, siendo ambas totalmente independientes. Generalmente se crean dos subconjuntos de características, con alguna excepción como la de Zhou & Li (2005), con tres subconjuntos, denominándolo como *tri-entrenamiento*.

Inicialmente se realiza un entrenamiento con la colección etiquetada para cada uno de los subconjuntos de características. Posteriormente, se clasifican las instancias de la colección no etiquetada, cuyos resultados más fiables se utilizan para entrenarse mutuamente los clasificadores. Así, lo que un clasificador aprende es transmitido a los otros, realizando el *co-entrenamiento*.

Este método requiere que todos los conjuntos de características utilizados sean adecuados para la clasificación, ya que en caso contrario el resultado no sería correcto.

Nigam & Ghani (2000) proponen una alternativa al co-entrenamiento, que trata de realizar una clasificación probabilística sobre la colección no etiquetada en lugar de basarse en los resultados más fiables. Esto se conoce también como *co-EM*.

Goldman & Zhou (2000) proponen también otra variante al co-entrenamiento, en el que utilizando dos clasificadores diferentes sobre todo el conjunto de características, se realiza un entrenamiento mutuo de la misma manera. Collins & Singer (1999) y Jones (2005), entre otros, han utilizado este tipo de técnicas.

De manera similar, el **aprendizaje multivista** o **multiview** tiene en cuenta la coincidencia entre clasificadores para el aprendizaje. Los prerrequisitos del co-entrenamiento no son necesarios, ya que son diferentes sistemas de aprendizaje inductivo los que se entrenan sobre la misma colección etiquetada, y deben ir coincidiendo en las predicciones realizadas sobre las instancias no etiquetadas. de˜Sa (1993) introdujo este sistema, aunque posteriormente se ha utilizado en múltiples ocasiones (Sindhwani *et al.* (2005b), Brefeld & Scheffer (2006)).

2.5.3.3. Modelos generativos

Los modelos generativos se basan en la suposición $p(x, y) = p(y) \cdot p(x|y)$, donde $p(x|y)$ es una distribución mixta identificable, entre los que los más utilizados son el Modelo Mixto de Gauss, Naïve Bayes o el Modelo Oculto de Markov; en los modelos discriminativos, por otro lado, se trabaja directamente sobre el cálculo de $p(y|x)$. Un único ejemplo de cada tipo, y un gran número de instancias sin etiquetar pueden ser suficientes para la aplicación de este modelo. Es un requisito importante que el modelo sea identificable (Ratsaby & Venkatesh (1995), Corduneanu & Jaakkola (2001)), ya que en caso contrario no es aplicable.

Nigam *et al.* (2000) y Fujino *et al.* (2008), entre otros, han utilizado técnicas basadas en modelos generativos.

Una conocida variante de este método es el denominado *cluster-and-label* (Demiriz *et al.* (1999), Dara *et al.* (2002)), que utiliza técnicas de clustering para agrupar la colección completa, etiquetándola posteriormente basándose en los ejemplos etiquetados.

2.5.3.4. Máquinas de Vectores de Soporte semisupervisadas (S^3VM)

Esta técnica basada en las máquinas de vectores de soporte (SVM) es conocida también como máquinas de vectores de soporte transductivas (TSVM), aunque debido a la naturaleza inductiva que sus variantes han adquirido en alguna ocasión, es más correcto el nombre de máquinas de vectores de soporte semisupervisadas (S^3VM), ya que el aprendizaje que realizan no tiene porqué ser transductivo.

Es, básicamente, una adaptación del método supervisado de las SVM hacia las técnicas de aprendizaje semisupervisado, donde la fase de aprendizaje se extiende a las instancias no etiquetadas. El objetivo es el de obtener unos límites de decisión en regiones poco pobladas, que maximicen los márgenes entre las áreas más pobladas, definidas por las instancias etiquetadas y no etiquetadas.

La función objetivo sigue siendo la misma que para las SVM supervisadas:

$$f(x) = \omega \cdot x + b$$

No obstante, se añade un parámetro adicional a la función simplificada empleada para la optimización:

$$\text{mín}\, \frac{1}{2} \cdot ||\omega||^2 + C \cdot \sum_{i=1}^{l} \xi_i^d + C^* \cdot \sum_{j=1}^{u} \xi_j^{*d}$$

Sujeto a:

$$y_{l_i}(\omega \cdot x_{l_i} + b) \geq 1 - \xi_i, \xi_i \geq 0$$

$$y^*_{u_j}(\omega \cdot x^*_{u_j} + b) \geq 1 - \xi^*_j, \xi^*_j \geq 0$$

2.5.3.5. Algoritmo de expectación-maximización (EM)

La idea básica del algoritmo de expectación-maximización (Dempster *et al.* (1977)) es entrenar un clasificador usando solo ejemplos etiquetados, usando posteriormente ese clasificador para asignar probabilidades-pesos de etiquetas a los ejemplos no etiquetados. Después se entrena un nuevo clasificador usando esas estimaciones y se vuelven a asignar pesos.

El algoritmo de expectación-maximización se puede realizar, por ejemplo, basándose en el método Naïve Bayes. Por lo tanto, dadas unas instancias etiquetadas y otras no etiquetadas, se crea un clasificador inicial con las no etiquetadas. Posteriormente es cuando se comienza a realizar una serie de iteraciones EM hasta su convergencia:

- **Paso E:** Se utiliza el clasificador para estimar la categorización de los documentos no etiquetados.
- **Paso M:** Basándose en todos los documentos disponibles (ya etiquetados), se redefinen los parámetros del clasificador.

Uno de los problemas que pueden surgir de la utilización de este algoritmo puede ser su convergencia hacia máximos locales. Uno de los motivos puede ser la disposición de pequeñas colecciones etiquetadas, como es

frecuente en aprendizaje semisupervisado, y su gran influencia hacia una convergencia incorrecta. Una de las soluciones empleadas para que esto no ocurra suele ser la asignación de un peso inferior a los documentos de la colección no etiquetada, por lo que los documentos ya etiquetados siempre tendrán un peso mayor. Otra de las variantes utilizadas es la extracción de componentes que determinan cada clase desde las instancias etiquetadas, clasificando después según las probabilidades de éstos.

Se pueden encontrar unos estudios recientes sobre este algoritmo en Prescher (2005) y Borman (2004).

2.5.3.6. Métodos basados en grafos

Los métodos basados en grafos definen un grafo en el que los nodos son instancias, tanto etiquetadas como no etiquetadas, y los vínculos que los conectan definen la similitud entre ellos, al que generalmente se asigna un peso que valora su similitud en cuanto a clasificación, aunque en algunos casos puede limitarse a la definición de valores binarios 0 y 1, mediante los cuales se define su igualdad o no, sin valorar el grado. Basándose en ello, son varios los métodos que se han utilizado hasta la actualidad, aunque el estudio realizado en la construcción de los grafos es aún muy reducido, tal y como explica Zhu (2005).

Los grafos se suelen representar en forma de matriz de pesos W de dimensión n x n, siendo n el número de instancias. Cada uno de los valores W_{ij} de esta matriz indica el peso de la unión entre ambas instancias, el cual será 0 para todo $i = j$.

Técnicas de regularización Muchas técnicas basadas en grafos se basan en la estimación de una función f para representar el grafo mediante la regularización del grafo, es decir, que cada nodo de éste tenga el mismo número de aristas en sí. Esta función debe aproximarse a las etiquetas definidas en los nodos etiquetados mediante una función de pérdida, y se debe conseguir que sea uniforme en todo el grafo mediante un regularizador.

Una de las ideas más utilizadas para la aproximación hacia esa función es la de conseguir que no existan demasiados saltos entre nodos, para que los caminos sean cortos. Para ello, frecuentemente se define el problema mediante un grafo Laplaciano. Esto se produce mediante la creación de la matriz diagonal D, cuyos valores D_{ii} corresponden a $\sum_{j=1}^{n} W_{ij}$. Para la creación de la matriz resultante entre W y D se opta por dos opciones diferentes:

- *Laplaciano no normalizado:* $L = D - W$
- *Laplaciano normalizado:* $\zeta = I - D^{-1/2} \cdot W \cdot D^{-1/2}$

Se han utilizado diversos métodos que definen una función de pérdida y un regularizador; entre ellos, los siguientes:

- **Mincut:** Este algoritmo fue propuesto por Blum & Chawla (2001). Únicamente permite la clasificación de las instancias de forma binaria, es decir, sobre dos categorías, como puede ser la de los casos positivos y negativos. Su idea principal es la de partir el grafo en unos determinados subgrafos en los que se minimiza la suma de los pesos de sus uniones, denominados mincut, lo cual minimiza el número de los pares de ejemplos similares que difieren en su categorización. Posteriormente, Blum *et al.* (2004) y Pang & Lee (2004) han utilizado también este algoritmo.

- **Campos Aleatorios Discretos de Markov:** Zhu & Ghahramani (2002) y Getz *et al.* (2005) han utilizado estrategias basadas en el cálculo de probabilidades marginales de los campos aleatorios discretos de Markov. Aunque puede resultar una buena opción, el problema de inferencia es muy complicado.

- **Consistencia Local y Global:** El método de consistencia local y global (Zhou *et al.* (2004)) utiliza la función de pérdida y el Laplaciano normalizado como regularizador.

- **Regularización de Tikhonov** (Bauer *et al.* (2007)).

- **Regularización de Manifold** (Belkin *et al.* (2006)).

Inducción en los métodos basados en grafos La mayoría de las técnicas basadas en grafos son de naturaleza transductiva, por lo que su aplicación a instancias que no se encuentran en la colección etiquetada y no etiquetada inicial puede resultar complicado. Aún así, recientemente se han realizado algunas propuestas para que estos métodos puedan ser inductivos y trabajar con nuevas instancias no vistas. La mayoría de las técnicas actuales se centran en la idea de mantener el grafo obtenido con la colección inicial de instancias etiquetadas y no etiquetadas, suponiendo que los ejemplos no vistos no alterarán la estructura del grafo. Esta suposición no tiene por qué ser real, pero se ahorra el gran coste computacional que supondría la reestructuración del grafo por cada nuevo ejemplo visto.

Algunas de las primeras propuestas con este objetivo fueron Zhu *et al.* (2003b), que proponían la utilización de técnicas de *k-NN* (ver sección 2.5.2.8) con los ejemplos no vistos, y Chapelle *et al.* (2003), que utiliza una combinación lineal de los ejemplos etiquetados y no etiquetados para la aproximación de las instancias no vistas. Delalleau *et al.* (2005) proponen la aplicación del método Nyström (Fowlkes *et al.* (2004)) para la clasificación de ejemplos no vistos:

$$f(x) = \frac{\sum_{i \in L \cup U} \omega_{xi} \cdot f(x_i)}{\sum_{i \in L \cup U} \omega_{xi}}$$

donde:

L es el número de documentos etiquetados (labeled), y

U es el número de documentos no etiquetados (unlabeled)

Por otra parte, Belkin *et al.* (2006) utilizan la regularización de grafos combinada con un algoritmo inductivo, en este caso SVM, con el fin de aprovechar las ventajas de cada uno. En trabajos derivados, Krishnapuram *et al.* (2005) se basan en la regresión logística para la regularización, mientras que Sindhwani *et al.* (2005a) presentan una técnica en la que el kernel del clasificador se define sobre todo el espacio, no únicamente sobre los ejemplos de entrenamiento.

Otras ideas de los métodos basados en grafos Existen otras ideas adicionales que se han aplicado a los métodos basados en grafos, con el fin de mejorar su rendimiento. Entre ellas está la de añadir enlaces entre nodos disimilares, además de los que comparten similitud, basándose en la idea de que eso es información adicional. La opción más sencilla podría ser la de asignar pesos negativos a los casos disimilares, pero esa no sería una buena opción, ya que los valores de la función no tendrían límites y ésta dejaría de ser convexa. Goldberg *et al.* (2007) definen una función que tiene en cuenta los casos disimilares. La siguiente función es la que define el peso de los enlaces del grafo:

$$\omega_{ij} \cdot (f(x_i) + f(x_j))^2$$

Así, en caso de que dos instancias sean disimilares, $f(x_i)$ y $f(x_j)$ tendrán signos opuestos con valores absolutos similares, lo que minimizará el resultado.

Otra de las ideas es la utilización, en algún caso, de grafos dirigidos. Por ejemplo, Zhou & Li (2005) realizan una aproximación a la filosofía de *hubs* y *authorities* del algoritmo HITS (Kleinberg (1999)) mediante grafos dirigidos, donde los primeros apuntan a los segundos, y Lu & Getoor (2003) utilizan también este tipo de representación para una estructura de enlaces de la Web.

2.5.3.7. Otros aspectos a tener en cuenta en aprendizaje semisupervisado

Además de todo lo mencionado anteriormente, existen otras ideas que, de forma general sobre diferentes algoritmos de aprendizaje semisupervisado,

se han aplicado en algunas ocasiones. Es el caso de la definición de proporciones por clases. Ante el posible problema de que el aprendizaje realizado sobre las instancias etiquetadas dé lugar a la clasificación de todas las no etiquetadas sobre la misma categoría, o que alguna de ellas quede discriminada, se han utilizado técnicas basadas en la definición de proporciones por clases, pudiendo establecer un determinado porcentaje de instancias para cada clase. Esta necesidad puede aparecer, sobretodo, cuando la separación entre clases es mínima. Con esta idea, Zhu *et al.* (2003a), por ejemplo, utilizan el procedimiento llamado *class mass normalization* (*CMN*), donde una instancia d_j pertenecerá a una clase si:

$$q \cdot \frac{f(d_j)}{\sum_j f(d_j)} > (1-q) \cdot \frac{1-f(d_j)}{\sum_j (1-f(d_j))}$$

siendo q el porcentaje de instancias que pertenecen a esa categoría entre las etiquetadas, y $f(d_j)$ el valor devuelto por la función de clasificación. Cumpliendo esta condición se consigue que cada categoría tenga un mínimo de instancias en sí.

En el caso del S^3VM, por ejemplo, no hace falta la definición de proporciones, ya que se realiza de forma explícita.

Otra de las ideas que alguna vez se ha utilizado en aprendizaje semisupervisado es la de utilizar las instancias no etiquetadas para la representación. En primer lugar, se utilizan las instancias no etiquetadas para aprender los rasgos que los caracterizan, para poder representar después las instancias etiquetadas mediante éstos. Como ejemplo, Ando & Zhang (2005) y Ando & Zhang (2007) presentan un sistema capaz de separar los rasgos de una colección en dos subconjuntos independientes entre sí, aunque a diferencia del *co-entrenamiento* (ver sección 2.5.3.2) éstos no tienen porqué ser suficientes para la clasificación por sí solos. A partir de ahí, crean un clasificador que predice la función de uno de los subconjuntos desde la perspectiva del otro.

2.6. Evaluación

La evaluación de un sistema de clasificación se suele basar, comúnmente, en los valores *TP* (positivos correctos), *FP* (falsos positivos), *TN* (negativos correctos) y *FN* (falsos negativos). No obstante, la importancia de cada uno de estos valores puede variar de forma significativa, por lo que su forma de evaluación puede cambiar. Por ejemplo, en un sistema de filtro de spam, adquiere mucha mayor importancia los *FN* (correo descartado que no es spam) que los *FP* (spam que no se ha considerado como tal). Asimismo, para la comparación de diferentes sistemas de clasificación, conviene que el cálculo de estos valores se realice sobre la misma colección de documentos.

2.6.1. Colecciones de documentos

Existen algunas colecciones de documentos que han sido las que más se han utilizado en la investigación entorno a la clasificación automática. El hecho de disponer de unas colecciones comunes y poder presentar unos análisis basados en los mismos documentos, hace que se puedan realizar comparativas, y valorar qué algoritmos ofrecen mejores resultados. Así, la colección de documentos más utilizada hasta la actualidad ha sido la Reuters-215781, una colección liberada por Reuters, que contiene noticias correctamente etiquetadas.

No obstante, existen otras colecciones de documentos que se está utilizando cada vez más. Son dos colecciones liberadas también por Reuters: Reuters Corpus Volume 12 (RCV1, con unas 810.000 noticias en inglés) y Reuters Corpus Volume 2 (RCV2, con unas 487.000 noticias en 13 idiomas). La primera se liberó en 2000 y la segunda en 2005, pero al ser su objetivo exclusivamente para investigación, actualmente hay que rellenar algunos formularios para solicitar las colecciones, después de que Reuters decidiera impedir su libre descarga en 2004.

2.6.1.1. Colecciones web

Son pocas las colecciones de datos de referencia existentes que estén basadas en páginas web. El hecho de que sea un área en el que se ha trabajado menos que en texto plano supone que las colecciones disponibles sean más reducidas. A continuación se mencionan algunas de las más extendidas:

- *BankSearch* (Sinka & Corne (2002)), compuesta por 10.000 páginas web sobre 10 clases: bancos comerciales, construcción, agencias aseguradoras, java, C, visual basic, astronomía, biología, fútbol y motociclismo.
- *WebKB* [2], formada por 4.518 documentos extraídos de 4 sitios universitarios y clasificados sobre 7 clases (estudiante, facultad, personal, departamento, curso, proyecto y miscelanea).
- *Yahoo! Science* (Tan *et al.* (2002)), que tiene 788 documentos científicos, clasificados sobre 6 ámbitos diferentes de la ciencia (agricultura, biología, ciencias terrestres, matemáticas, química y otros).

2.6.2. Métodos de evaluación

Se han utilizado diferentes técnicas a la hora de evaluar la calidad de un sistema de clasificación automática, debido a que en cada uno de los casos el objetivo que se busca es diferente. Las más conocidas son la *efectividad*, la *eficiencia* y la *utilidad*, que se detallan a continuación.

[2] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/

2.6.2.1. Efectividad

Los valores más utilizados para el cálculo de la efectividad de un sistema de clasificación son la precisión y la cobertura, aunque en algunas ocasiones pueden resultar útiles otros valores como la exactitud o el error, que se mencionan a continuación.

Precisión y cobertura (Medida F) La técnica más utilizada para la evaluación de la efectividad de diferentes clasificadores es la denominada F_β, o más concretamente F_1. Esta aproximación se basa en el cálculo de la precisión (π) y la cobertura (ρ):

$$\pi = \frac{TP}{TP + FP}$$

$$\rho = \frac{TP}{TP + FN}$$

Cabe destacar, sin embargo, que el cálculo de la precisión y la cobertura se puede realizar de dos maneras diferentes:

- **Micromedia (Microaveraging):** los cálculos se basan en la suma de todos los valores individuales.
- **Macromedia (Macroaveraging):** los cálculos se basan en la suma local de los valores para cada categoría, obteniendo después la media total.

El cálculo de F_β se basa en la asignación de pesos a los valores de la precisión y la cobertura:

$$F_\beta = \frac{(\beta^2 + 1) \cdot \pi \cdot \rho}{\beta^2 \pi + \rho}$$

El caso más habitual suele ser cuando se asigna el valor 1 a β, obteniendo F_1:

$$F_1 = \frac{2 \cdot \pi \cdot \rho}{\pi + \rho}$$

Generalmente se obtiene un valor óptimo de F_1 cuando la precisión y la cobertura tienen un valor parecido, pero muchas veces se puede complicar el cálculo de este punto óptimo.

Exactitud y error La exactitud ($\hat{A}$) y el error ($\hat{E}$) son dos valores relacionados entre sí, que aunque son frecuentemente utilizados en el área del aprendizaje automático, no lo son tanto en el entorno de la clasificación automática. Yang (1999) explica que habitualmente no son apropiados estos valores en la clasificación automática, debido al gran valor que adquiere el denominador de la división, haciendo que las diferencias en los resultados no sean significativas en muchos casos. El cálculo se realiza de la siguiente manera:

$$\hat{A} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\hat{E} = 1 - \hat{A} = \frac{FP + FN}{TP + TN + FP + FN}$$

2.6.2.2. Eficiencia

La efectividad es, con diferencia, el valor de evaluación más utilizado. La eficiencia es un valor que se calcula con mucha menor frecuencia, aunque en algunos casos puede servir para diferenciar entre sistemas con una efectividad similar. Para calcular la eficiencia debe tenerse en cuenta la eficiencia de entrenamiento y la eficiencia de clasificación.

2.6.2.3. Utilidad

La utilidad es un valor que trata de medir desde el aspecto económico las ganancias o pérdidas que un sistema puede suponer. Para ello, los valores de las ganancias u_{TP}, u_{FP}, u_{FN} y u_{TN} para uno de los casos deben ser ponderados de manera relativa al sistema específico, para así obtener su utilidad.

Capítulo 3

Experimentos

Este capítulo presenta el estudio comparativo realizado en este trabajo. Para comenzar, se exponen los motivos que han llevado a la decisión de utilizar SVM como técnica de clasificación, al margen de otras aproximaciones. Posteriormente, se detalla la configuración y características de la experimentación llevada a cabo, con el fin de comparar diferentes aproximaciones a la resolución del problema de clasificación de páginas web como tarea semisupervisada multiclase. Para finalizar, se muestran y analizan los resultados obtenidos tras la realización de la experimentación.

3.1. ¿Por qué SVM?

Las técnicas basadas en SVM están dando lugar al desarrollo de múltiples estudios sobre clasificación de textos, tal y como lo muestra la literatura en estos últimos años (Bolelli *et al.* (2007); Bordes *et al.* (2007); Sun *et al.* (2007); Wang *et al.* (2007a,b); Zien *et al.* (2007); Heymann *et al.* (2008)). Asimismo, debido al gran interés que ha creado, son muchas las implementaciones que se pueden encontrar de las variantes de este algoritmo[1].

SVM se basa en un modelo espacio vectorial (VSM) al igual que muchas otras técnicas de aprendizaje, pero aporta algunas ventajas sobre las demás, tal y como explica Joachims (1998):

- No se requiere una selección o reducción de términos. En caso de que una clase se distribuya en áreas separadas del espacio vectorial, será la redimensión mediante la función de kernel la que se ocupe de solucionarlo.
- No es necesario realizar un esfuerzo de ajuste de parámetros en el caso de problemas linealmente separables, ya que dispone de su propio método para ello.

[1] http://www.svms.org/software.html

Otra de las ventajas que ofrece SVM es que su transformación a aprendizaje semisupervisado se convierte, generalmente, en un comportamiento transductivo, lo que posibilita el máximo refinamiento en la definición del clasificador.

Por tanto, SVM fue la opción escogida para este trabajo, considerando estas y otras ventajas que ofrece. Además, más adelante se muestran los resultados de la experimentación comparativa que se ha llevado a cabo en este trabajo sobre otras técnicas de clasificación, lo cual muestra su gran efectividad ante este tipo de problemas.

No obstante, la naturaleza propia de SVM se limita a la clasificación binaria supervisada, lo que hace interesante el estudio de llevarlo más allá, tal y como se explica a continuación.

3.2. Problemática de SVM

En la última década, SVM se ha convertido en una de las técnicas más utilizadas para clasificación automática, debido a los buenos resultados que se han obtenido. Esta técnica se basa en la representación de los documentos en un modelo espacio vectorial, y asume que los documentos de cada clase son separables en el espacio de representación; en base a ello, trata de buscar un hiperplano que separe ambas clases. Entre todos los hiperplanos posibles que separan las clases, SVM se queda con aquélla que maximiza la distancia entre los documentos de cada clase y el propio hiperplano, lo que se denomina margen.

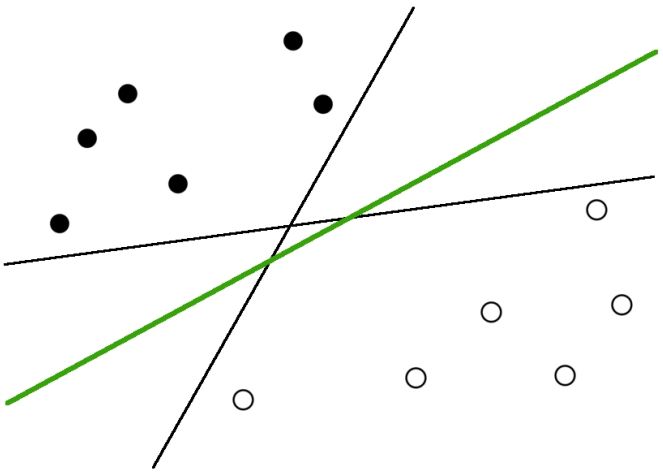

Figura 3.1: Selección de hiperplano que maximiza el margen con SVM

Este hiperplano se define mediante la siguiente función:

$$f(x) = w \cdot x + b$$

En caso de que el conjunto de datos sea separable, la función anterior obtendrá una correcta clasificación cuando $y_i(w \cdot x_i + b) > 0 \quad \forall i$. Esta relación, además, puede ser reescalada con el fin de obtener su forma canónica, de tal forma que $w \cdot x + b = 1$ defina el hiperplano de margen de una clase, y $w \cdot x + b = -1$ defina las de la otra.

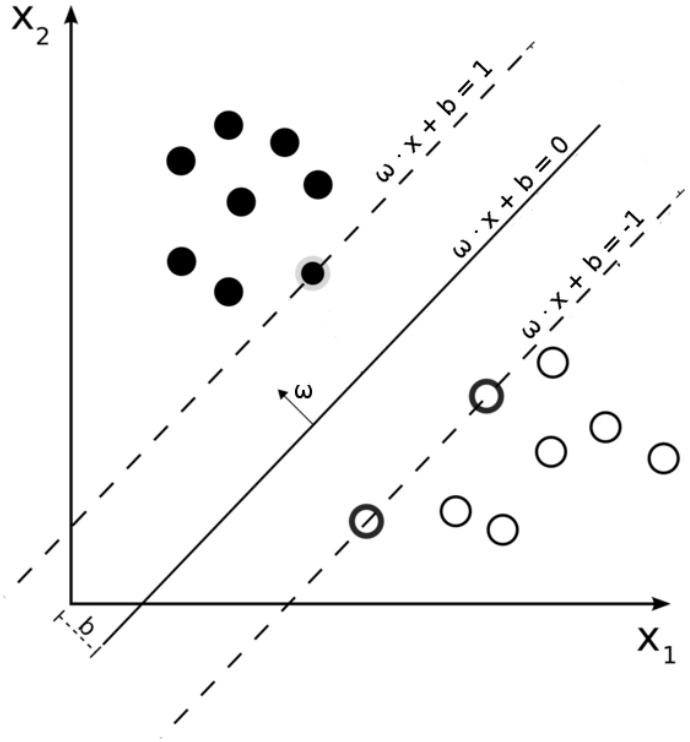

Figura 3.2: Función de clasificación para SVM

Esta función es la que define el hiperplano de separación entre las clases. No obstante, esto supondría tener en cuenta todos los valores posibles para la función, para después quedarse con los que maximicen los márgenes, lo que resulta muy difícil de optimizar, y se complica aún más cuando el conjunto de datos no es fácilmente separable, por lo que se utiliza la siguiente función de optimización equivalente (Boser *et al.* (1992); Cortes & Vapnik (1995)):

$$\text{mín}\, \frac{1}{2}||w||^2 + C\sum_{i=1}^{l} \xi_i^d$$

$$\text{Sujeto a:} \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i, \xi_i \geq 0$$

donde:

C es el parámetro de penalización,

ξ_i es la distancia entre el hiperplano y el documento i, y

d tiene un valor de 1 para un coste lineal, y un valor de 2 para un coste cuadrático

La utilización de esta función sí que resulta óptima, y el coste computacional se reduce de forma considerable. El inconveniente que plantea es que de esta manera únicamente se resuelven problemas linealmente separables, por lo que en muchos casos se requiere de la utilización de una función de kernel para la redimensión del espacio; de este modo, el nuevo espacio obtenido sí que resulta linealmente separable. Posteriormente, la redimensión se deshace, de modo que el hiperplano encontrado será transformado al espacio original, constituyendo la función de clasificación. Las funciones de kernel más conocidas utilizadas hasta la actualidad son las siguientes(Campbell (2000)):

- Lineal:
$$K(x_i, x_j) = x_i \cdot x_j$$
- Polinomial:
$$K(x_i, x_j) = (\gamma \cdot x_i^T \cdot x_j + r)^d, \gamma > 0$$
- Radial Basis Function (RBF):
$$K(x_i, x_j) = e^{(-\gamma \cdot ||x_i - x_j||^2)}$$
- Sigmoidea:
$$K(x_i, x_j) = \tanh(\gamma \cdot x_i^T \cdot x_j + r)$$

Es importante destacar que esta función únicamente puede resolver problemas binarios, ya que el hiperplano obtenido sólo es capaz de separar dos subespacios.

3.2.1. SVM multiclase

Debido a la naturaleza dicotómica de SVM, surgió la necesidad de implementar nuevos métodos que pudieran resolver problemas multiclase. Con este objetivo, se han propuesto diferentes aproximaciones. Por una parte, como aproximación directa, Weston & Watkins (1998) proponen una modificación de la función de optimización que tiene en cuenta todas las clases:

$$\text{mín}\, \frac{1}{2} \sum_{m=1}^{n} ||w_m||^2 + C \sum_{i=1}^{l} \sum_{m \neq y_i} \xi_i^m$$

Sujeto a:

$$w_{y_i} \cdot x_i + b_{y_i} \geq w_m \cdot x_i + b_m + 2 - \xi_i^m, \xi_i^m \geq 0$$

Por otra parte, diversas técnicas para la aproximación a SVM multiclase de k clases se han basado en la combinación de clasificadores binarios (Hsu & Lin (2001)):

- *one-against-all* construye k clasificadores que definen otros tantos hiperplanos que separan la clase i de los *k-1* restantes. Por ejemplo, para un problema de 4 clases, se crean los clasificadores *1 vs 2-3-4*, *2 vs 1-3-4*, *3 vs 1-2-4* y *4 vs 1-2-3*. Al recibir nuevos documentos, éstos son sometidos a los k clasificadores, escogiendo como resulta aquélla que maximiza el margen:

$$\hat{C}_i = \arg \max_{i=1,\ldots,k} (w_i x + b_i)$$

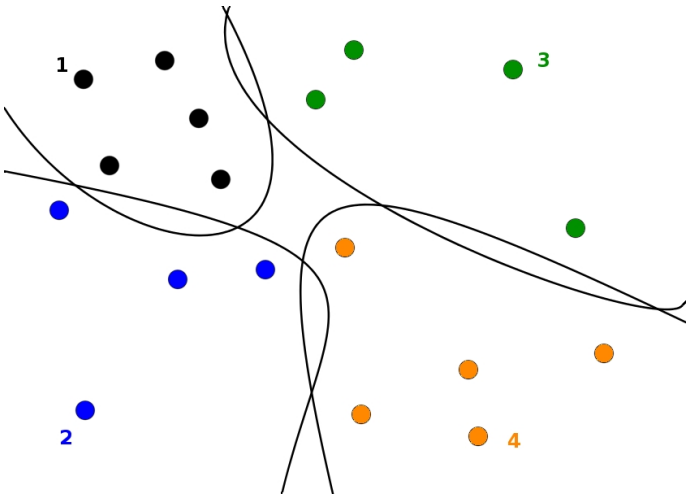

Figura 3.3: Clasificación one-against-all

- *one-against-one* construye $\frac{k(k-1)}{2}$ clasificadores, uno para cada par de clases posible, enfrentando así a todas las clases una a una. Por ejemplo, un problema con 4 clases generaría los siguientes clasificadores: *1 vs 2*, *1 vs 3*, *1 vs 4*, *2 vs 3*, *2 vs 4* y *3 vs 4*. Una vez realizado esto, se somete a cada documento de la colección de test a todos estos clasificadores, donde se añade un voto a la clase ganadora para cada caso. Finalmente, aquélla que más votos obtenga será la clase propuesta por el sistema.

Existen otras aproximaciones derivadas, entre las que destaca la denominada *Directly Acyclic Graph SVM (DAGSVM)* (Platt *et al.* (2000)), que

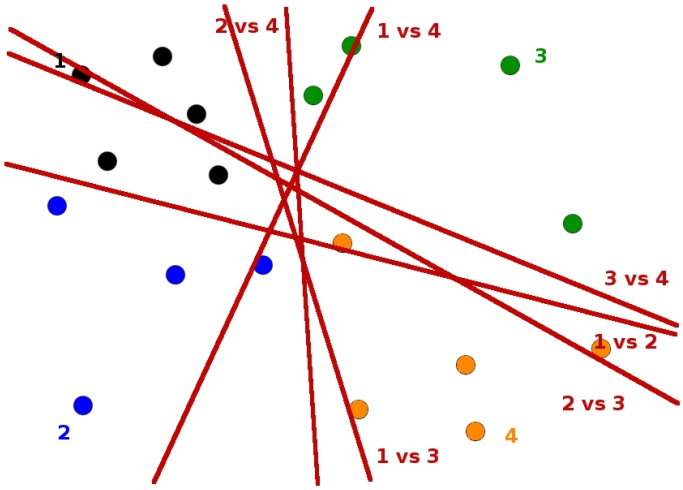

Figura 3.4: Clasificación one-against-one

se basa en la aproximación *one-against-one*, aunque modifica el proceso de decisión durante la fase de clasificación. En ella, se crea un grafo en forma de árbol con $\frac{k(k-1)}{2}$ nodos, donde k nodos son finales. Recorriendo los nodos de este árbol se decide la clase a la que pertenece la instancia.

3.2.2. Aprendizaje semisupervisado para SVM (S^3VM)

Las técnicas basadas en aprendizaje semisupervisado se diferencian en la fase de aprendizaje, ya que hacen uso también de las predicciones realizadas sobre documentos no etiquetados además de los documentos etiquetados que ya se utilizan en aprendizaje supervisado Joachims (1999). Tener en cuenta los documentos no etiquetados puede mejorar los resultados de la tarea de clasificación, como muestra la figura 3.5, sobre todo cuando el número de documentos etiquetados del que se dispone es muy reducido respecto a la colección. Las SVM semisupervisadas se conocen también por las siglas S^3VM (Semi-Supervised Support Vector Machines).

Las S^3VM pueden ser de naturaleza tanto inductiva como transductiva (ver sección 2.5.3). Inicialmente estas técnicas eran conocidas como las máquinas de vectores de soporte transductivas (TSVM), aunque su aplicación inductiva en algunos casos hizo que se comentara a utilizar el nombre de S^3VM, al ser más correcto. Al enfrentar ambas técnicas entre sí, se mostró que las técnicas transductivas superan a las inductivas (Demiriz & Bennett (2000)).

La transformación de SVM a aprendizaje semisupervisado requiere la inclusión de un nuevo término en la función de optimización:

$$\text{mín}\,\frac{1}{2}\cdot||\omega||^2 + C\cdot\sum_{i=1}^{l}\xi_i^d + C^*\cdot\sum_{j=1}^{u}\xi_j^{*^d}$$

El objetivo que busca SVM se basa en una función convexa, mientras que el de S^{3}VM se define con una función no convexa, lo que hace que sea un problema NP-duro. Ésta puede disponer de varios mínimos locales, lo que puede dar lugar a la obtención de un resultado que no sea el óptimo. Ante esto, se ha trabajado en la búsqueda de soluciones eficientes que se aproximaran. Joachims (1999) fue de los primeros en presentar una solución basada en este método, con el software SVM-light, aunque posteriormente son muchos los que han seguido trabajando con este tipo de técnicas, y son múltiples las aproximaciones propuestas para relajar el problema, destacando las que se mencionan a continuación:

- **$\bigtriangledown$S3VM:** Chapelle & Zien (2005) proponen esta técnica que utiliza el algoritmo de descenso de gradiente para buscar los límites para cada clase. Estos límites evitan las regiones de alta densidad, mediante la búsqueda de mínimos locales. Este método se basa en la suposición de clusters, lo que significa que un punto perteneciente a un cluster únicamente corresponde a una clase.

- **Continuation method:** Esta técnica propuesta por Chapelle *et al.* (2006b) utiliza algoritmos de suavizado (smoothing) para la simplificación de la función no convexa y convertirla en convexa, después de varias iteraciones de suavizado. El suavizado consta, básicamente, de unos algoritmos de obtención de patrones importantes desde los datos, dejando a un lado el ruido.

- **Concave-Convex Procedure (CCCP):** (Collobert *et al.* , 2006) proponen la optimización de la función no convexa mediante la utilización del Concave-Convex Procedure (CCCP) (Yuille & Rangarajan (2003)). Esta técnica asume que dicha función puede ser representada como la suma de su parte convexa y de su parte cóncava; se realizan varias iteraciones de este proceso hasta su convergencia.

- **Ramificación y Poda (Branch and Bound):** Chapelle *et al.* (2006a) proponen la integración de esta técnica sobre S^{3}VM. El algoritmo clásico de Ramificación y Poda enumera inicialmente todas las soluciones candidatas (ramificación), eliminando posteriormente las no óptimas mediante la utilización de límites superiores e inferiores que optimicen el resultado (poda). Su aplicación a las S^{3}VM se aplica de la siguiente manera, con el fin de minimizar la función f representada sobre el espacio X:

- *Ramificación:* el espacio X se divide de forma recursiva en subespacios más pequeños, con lo que resulta una estructura en forma de árbol donde cada nodo es un subespacio.
- *Poda:* se procede a la eliminación de subespacios. Para ello, se escogen dos subespacios A y $B \subset X$. Suponiendo que se conoce un límite superior (en adelante a) que es el valor más alto de la función f para el espacio A, y se conoce un límite inferior (en adelante b) que es el valor más bajo de la función f para el espacio B, y se obtiene que $a < b$, se sabe que hay un elemento en A con menor valor que cualquier elemento de B. Por tanto, en la búsqueda del mínimo global, se pueden eliminar los elementos de B; en consecuencia, se poda el subespacio B. Este proceso se repite hasta la obtención del mínimo local.

Esta técnica ofrece una exactitud alta sobre pequeñas colecciones, aunque tal y como sus autores comentan puede no resultar útil para grandes colecciones.

- **Recocido determinístico (Deterministic annealing):** Sindhwani *et al.* (2006) proponen esta técnica basada en la homotopía, es decir, en la deformación de la función inicial mapeándolo sobre un espacio más reducido. El algoritmo realiza dos pasos:
  - En primer lugar, las variables discretas son tratadas como variables aleatorias, sobre las que se define un espacio de distribuciones probabilísticas.
  - Posteriormente, se reemplaza la función original por una función continua optimizada sobre el nuevo espacio definido que minimiza el resultado de la función objetivo. Esta optimización se realiza mediante la utilización de la homotopía.
- **Programación semidefinida (Semi-definite programming):** Bie & Cristianini (2004, 2006) y Xu & Schuurmans (2005) han utilizado técnicas de programación semidefinida basadas en matrices semidefinidas positivas, cuyo mayor inconveniente es el gran coste computacional que suponen.

De todas maneras, la mayoría del trabajo realizado sobre S^3VM se ha centrado en clasificación binaria, y apenas se ha investigado en su aplicación a problemas de clasificación multiclase.

3.2.3. S^{3}VM multiclase

En los problemas donde la taxonomía dispone de más de dos categorías y el número de documentos previamente etiquetados es muy pequeño, se

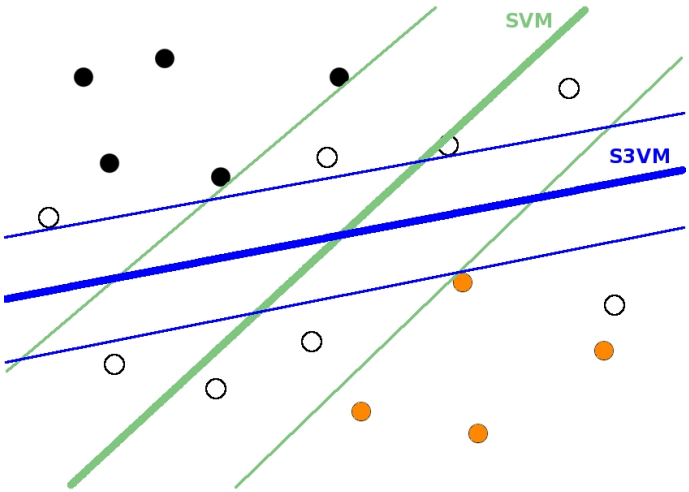

Figura 3.5: SVM vs S^{3}VM, donde los círculos blancos representan documentos no etiquetados.

precisa la combinación de las dos técnicas anteriormente expuestas, lo que supone un método de S^{3}VM multiclase. Los problemas reales de clasificación de páginas web suelen cumplir con estas características, ya que el número de categorías suele ser mayor que dos, y la pequeña colección de documentos etiquetados de la que se dispone normalmente implica la necesidad de utilizar documentos no clasificados en la fase de entrenamiento.

La única aproximación a S^{3}VM multiclase encontrada en la literatura ha sido propuesta por Yajima & Kuo (2006), con una técnica que traslada la función multiclase directa al entorno semisupervisado. La función de optimización resultante es la siguiente:

$$\text{mín}(\frac{1}{2}\sum_{i=1}^{h}\beta^{i^T}K^{-1}\beta^i + C\sum_{j=1}^{l}\sum_{i\neq y_j}\text{máx}\{0, 1-(\beta_j^{y_j}-\beta_j^i)\}^2)$$

donde β representa el producto entre un vector de variables y una matriz de kernel definidas por el autor.

Esta función de optimización, sin embargo, puede resultar muy costosa, debido a la cantidad de variables que se deben tener en cuenta en el proceso de minimización de la misma, lo que hace interesante el problema de encontrar otros enfoques a S^{3}VM multiclase.

3.3. Propuestas para S^{3}VM multiclase

Ante la escasez de propuestas para la implementación de métodos de S^{3}VM multiclase, nuestro objetivo es el de proponer y comparar diversas

técnicas aplicables a este entorno, basándose en las ya utilizadas para problemas supervisados multiclase y semisupervisados binarios.

- **2-steps-SVM**: Hemos denominado así a la técnica que se basa en la aproximación supervisada multiclase explicada en la sección 3.2.1. Este método trabaja, en el primer paso, sobre la colección de entrenamiento, aprendiendo con los documentos etiquetados y prediciendo los no etiquetados; a posteriori, se etiquetan estos últimos según las predicciones obtenidas. Como segundo paso, se realiza la clasificación habitual para este método, ya que ahora la colección se ha convertido en supervisada, con todos los ejemplos de entrenamiento etiquetados.

- **one-against-all-S^{3}VM** y **one-against-one-S^{3}VM** son propuestas basadas en la combinación de clasificadores binarios semisupervisados, que aunque se han utilizado en colecciones supervisadas, nunca han sido experimentadas sobre colecciones con documentos no etiquetados. Sin embargo, el enfoque *one-against-one-S^{3}VM* plantea un problema intrínseco de ruido en la fase de entrenamiento con los documentos no etiquetados, ya que cada clasificador para un par de categorías únicamente debe ser alimentado por documentos que le correspondan, y el problema radica en la imposibilidad de excluir aquellos ejemplos no etiquetados que no deberían incluirse.

- **all-against-all-S^{3}VM**: Aparte de los dos anteriores, en este trabajo se presenta una nueva propuesta de combinación de clasificadores binarios, que hemos denominado *all-against-all-S^{3}VM*, y que podría ser utilizada tanto para aprendizaje supervisado como semisupervisado. En ella se definen $2^{n-1}-1$ clasificadores, correspondientes a todos los enfrentamientos posibles entre las clases, teniendo en cuenta que todas las clases deben caer en uno u otro lado. Por ejemplo, para un problema de cuatro clases, se generarán los clasificadores *1 vs 2-3-4*, *1-2 vs 3-4*, *1-2-3 vs 4*, *1-3 vs 2-4*, *1-4 vs 2-3*, *1-2-4 vs 3* y *1-3-4 vs 2*.

  Cada nuevo documento recibido en la fase de clasificación se someterá a cada uno de los clasificadores generados. Una vez obtenidos esos resultados, el sistema debe predecir la clase a la que asigna cada documento. Para ello, se probaron y compararon cuatro métodos diferentes de decisión, los cuales se basan en diferentes criterios de votación. Estos sistemas de votación se centran en las predicciones de los diferentes clasificadores y, en base a ello, premian a aquellas clases que quedan en la parte ganadora del clasificador con cierta puntuación. Esta puntuación puede ser diferente en base al criterio de votación utilizado:

  - **Votación simple:** El voto añadido a las clases ganadoras tiene

un valor de 1:

$$V_{C_i} = V_{C_i} + 1$$

- **Votación por margen:** Se añade un voto que tiene un valor equivalente al margen positivo ($|\xi|$) entre el documento y el hiperplano de separación, por lo que cuanto más lejos estén supone una clasificación más clara.

$$V_{C_i} = V_{C_i} + |\xi|$$

- **Votación por rivalidad:** El valor del voto que se añade varía en función de la rivalidad existente en el clasificador. Es decir, tiene un mayor peso que sobre el clasificador *1 vs 2-3-4* gane la clase 1, que cualquiera de las tres restantes, por la importancia que tiene que gane el sólo y por la mayor probabilidad que hay, generalmente, de que caiga en la parte restante.

$$V_{C_i} = V_{C_i} + \frac{k - k_w}{k}$$

  siendo k el número total de clases, y k_w el número de clases que están en la parte ganadora.

- **Votación combinada:** Se añade un voto que tiene el valor del producto entre el margen y la rivalidad.

$$V_{C_i} = V_{C_i} + |\xi| \cdot \frac{k - k_w}{k}$$

Estos criterios de votación ofrecen una puntuación final para cada una de las clases, tras lo que el sistema se queda con aquélla que tiene la mayor puntuación, siendo esta la clase predicha.

$$\hat{C} = \text{máx}\, V_{C_i}, \forall i \in [1, .., k]$$

Tras múltiples pruebas sobre las colecciones utilizadas, se comprobó que la *votación simple* era superada por las otras tres técnicas, por lo que fue la primera en descartarse. Entre las tres técnicas restantes, se obtuvieron unos resultados notablemente superiores para el sistema de *votación por margen*, por lo que se optó por seguir utilizando este criterio de decisión.

3.4. Experimentación

Para la realización de la experimentación se ha procedido a la implementación de los algoritmos descritos en el apartado anterior, y su ejecución sobre las colecciones de datos escogidas. Todos los documentos de las

colecciones utilizadas están etiquetados, por lo que cada una de ellas se ha dividido en una colección de entrenamiento, que sirve para que el clasificador aprenda, y otra de test, que sirve para que el sistema cree las predicciones y se pueda evaluar su rendimiento. A continuación se explican con más detalle las características de la experimentación llevada a cabo.

3.4.1. Colecciones de datos

Para esta experimentación se han utilizado colecciones de páginas web de referencia que ya han sido probadas anteriormente en problemas de clasificación:

- *BankSearch* (Sinka & Corne (2002)), compuesta por 10.000 páginas web sobre 10 clases, de muy diversos temas: bancos comerciales, construcción, agencias aseguradoras, java, C, visual basic, astronomía, biología, fútbol y motociclismo. 4.000 ejemplos han sido asignados a la colección de entrenamiento, y los 6.000 restantes a la de test.

- *WebKB* [2], formada por 4.518 documentos extraídos de 4 sitios universitarios y clasificados sobre 7 clases (estudiante, facultad, personal, departamento, curso, proyecto y miscelánea). La clase miscelánea se ha eliminado de la colección debido a la ambigüedad, resultando 6 categorías. De todos los ejemplos que componen la colección, 2.000 se han asignado al entrenamiento y 2.518 al de test.

- *Yahoo! Science* (Tan *et al.* (2002)), que tiene 788 documentos científicos, clasificados sobre 6 ámbitos diferentes de la ciencia (agricultura, biología, ciencias terrestres, matemáticas, química y otros). Se han definido 200 documentos para el entrenamiento, y 588 para el test.

Desde la colección de entrenamiento, para cada caso, se han creado diferentes versiones, entre las que varía el número de documentos etiquetados, dejando el resto como no etiquetados, pudiendo probar así las diferentes aproximaciones semisupervisadas.

Para la representación vectorial de los documentos que componen cada colección, se ha basado en los valores tf-idf de los unitérminos encontrados en los textos, excluyendo los de mayor y menor frecuencia. Los unitérminos resultantes han sido los que han definido las dimensiones del espacio vectorial.

Cabe destacar las diferencias existentes entre las colecciones utilizadas, ya que los temas contenidos en ellos y su organización varía de forma considerable. La colección *BankSearch* es la más heterogénea de todas, ya que sus categorías son de ámbitos muy diferentes, y su clasificación puede resultar, por tanto, más sencilla. En el otro extremo está *WebKB*, una colección

[2] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/

muy homogénea, al tratarse de documentos de carácter académico, por lo que la similitud entre las categorías es mucho mayor. En el medio quedaría *Yahoo! Science*, que aunque todas sus categorías son sobre ciencia, abarcan diferentes ramas de ésta.

3.4.2. Implementación de los métodos

Para la implementación de los diferentes métodos de clasificación descritos en la sección 3.3, se requieren un clasificador semisupervisado binario y otro supervisado multiclase, para después combinarlos. Para el primero, se ha escogido SVMlight[3], y para el segundo, su derivado SVMmulticlass. Basándose en ambos algoritmos, se han implementado los correspondientes métodos para el comportamiento *2-steps-SVM* supervisado y las técnicas *one-against-all-S^3VM*, *one-against-one-S^3VM* y *all-against-all-S^3VM* semisupervisadas. Todos ellos están basados en aprendizaje transductivo (ver sección 2.5.3), por lo que utilizan los documentos de la colección de test para seguir refinando la función de clasificación.

Finalmente, además de los algoritmos comentados, se ha simplificado el algoritmo *2-steps-SVM* a un solo paso, *1-step-SVM*, donde utilizando únicamente un clasificador supervisado multiclase se entrena con los ejemplos etiquetados y se predicen los ejemplos de test, ignorando por tanto los ejemplos no etiquetados. Este método sirve para evaluar la aportación de los documentos no etiquetados en el aprendizaje.

3.4.3. Medidas de evaluación

La medida de evaluación escogida para el rendimiento de los algoritmos propuestos ha sido el "acierto" (accuracy), que es la que se viene utilizando en el área de la clasificación automática de textos. El acierto mide el porcentaje que comprende el número de predicciones correctas sobre el total de documentos testeados.

Se han considerado de la misma manera los aciertos sobre cualquiera de las clases, sin que ninguna de ellas tenga una mayor importancia respecto a las demás, por lo que no existe ponderación alguna en la evaluación.

3.5. Resultados

En primer lugar, con el fin de escoger una técnica semisupervisada que fuera adecuada para la clasificación automática de páginas web, se procedió a realizar una comparativa entre diferentes metódos. Para ello, debido a que no se disponía de ninguna técnica para la experimentación de SVM en un entorno semisupervisado multiclase, se implementó la primera aproximación propuesta para este propósito, la denominada *2-steps-SVM*. Además de

[3]http://svmlight.joachims.org

éste, se tuvieron en cuenta un algoritmo de *bootstrapping*, y otro basado en métodos probabilísticos de *Naive Bayes*.

Los resultados obtenidos para esta comparativa se muestran en las figuras 3.6, 3.7 y 3.8. Éstas gráficas muestran de forma clara la superioridad de SVM frente a las otras dos técnicas seleccionadas. Por tanto, se optó por la utilización de técnicas basadas en SVM para la resolución del problema, descartando las otras opciones.

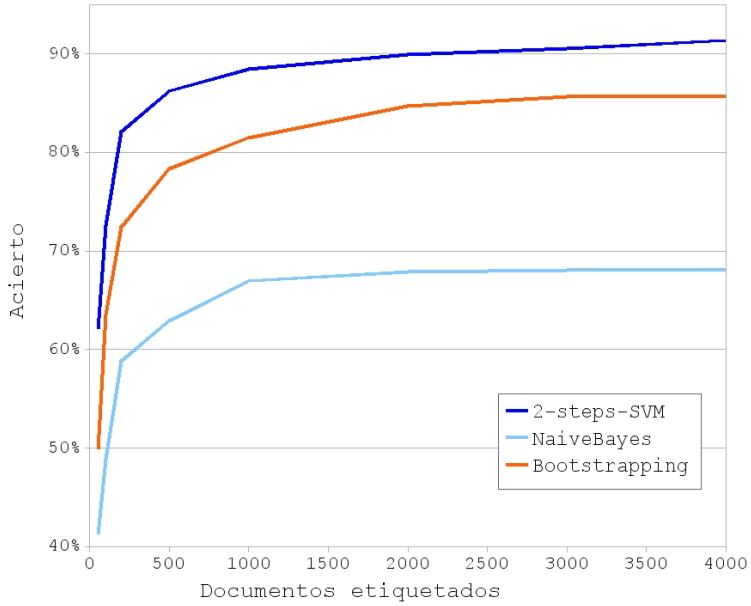

Figura 3.6: Resultados para BankSearch: Comparativa de técnicas semisupervisadas

En cuanto a la comparativa entre las técnicas propuestas para la transformación de SVM en multiclase y semisupervisado, en las figuras 3.9, 3.10 y 3.11 se muestran los resultados de la clasificación para las colecciones *BankSearch*, *WebKB* y *Yahoo! Science*, respectivamente, en función del tamaño de la muestra etiquetada. Cabe destacar que para cada una de estas muestras se realizaron 9 ejecuciones, y se obtuvo la media de todas ellas, que es la que se representa en las gráficas.

Los resultados obtenidos tras el análisis de las gráficas pueden resumirse en las siguientes ideas:

- En todos los casos el mejor comportamiento se obtiene para uno de los algoritmos basados en clasificadores multiclase supervisados, bien sea el *1-step-SVM* o el *2-steps-SVM*; incluso en los casos con menos

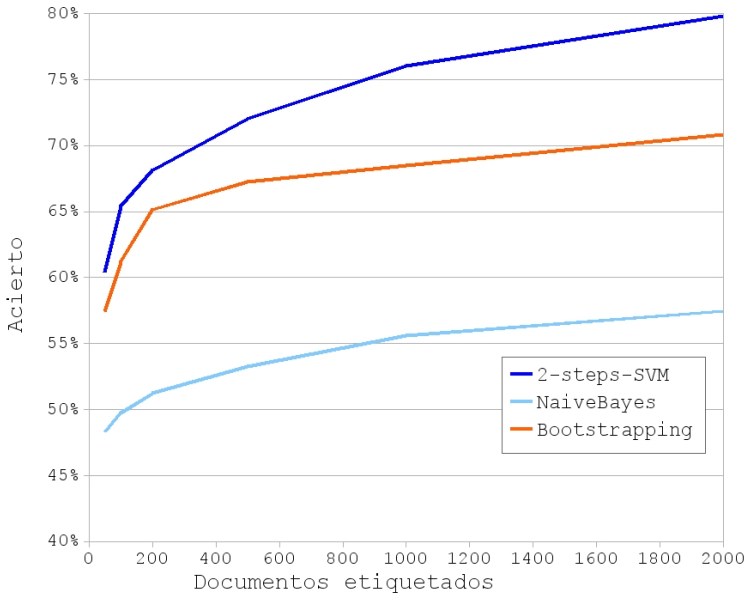

Figura 3.7: Resultados para WebKB: Comparativa de técnicas semisupervisadas

documentos etiquetados, destacan sobre los basados en clasificadores semisupervisados binarios.

- De las tres técnicas semisupervisadas comparadas, destaca la propuesta *all-against-all-S^3 VM*, ligeramente superior a *one-against-all-S^3 VM*, y muy superior a *one-against-one-S^3 VM*. Este último método, de hecho, demuestra que el ruido que se había previsto sí que existe, y que la calidad de los resultados obtenidos es baja.

- El método *1-step-SVM*, que ignora los documentos no etiquetados para la fase de aprendizaje, muestra unos resultados similares a los de *2-steps-SVM* para las colecciones *BankSearch* y *Yahoo! Science*, pero notablemente superiores para *WebKB*, donde las clases son más homogéneas. En este caso es donde mejor resulta ignorar los documentos no etiquetados, mediante el método *1-step-SVM*.

- Para todas las colecciones, según se aumenta el número de documentos etiquetados, se mantiene el ranking obtenido por los algoritmos.

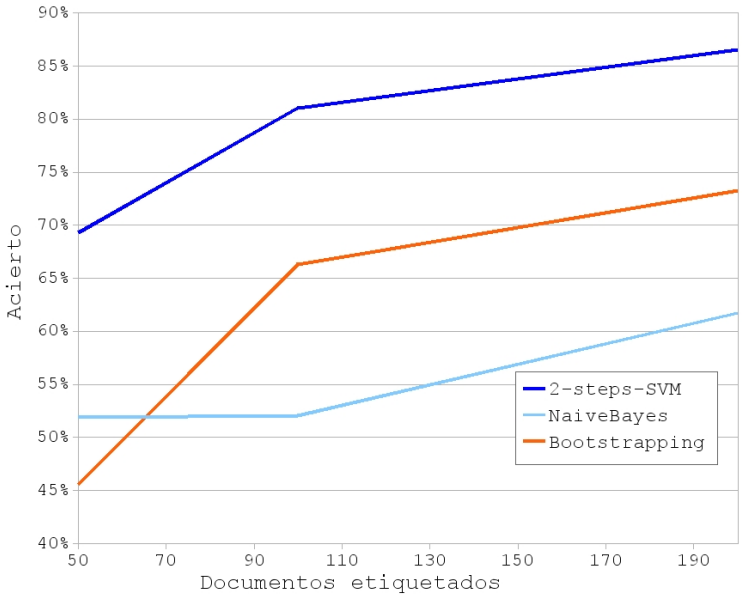

Figura 3.8: Resultados para Yahoo! Science: Comparativa de técnicas semi-supervisadas

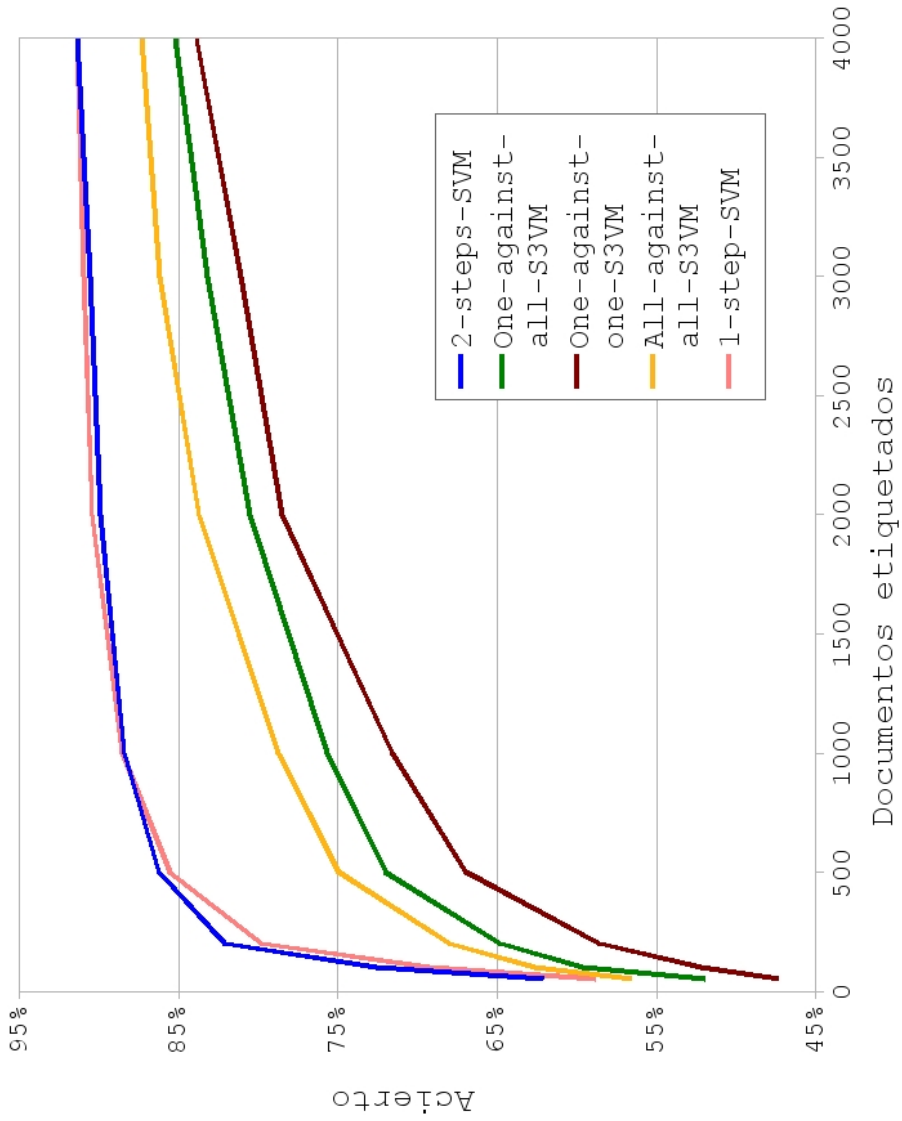

Figura 3.9: Resultados para BankSearch: Comparativa de técnicas S^3VM multiclase

núm. etiquetados \ algoritmo	2-steps	one-against-all-s3vm	one-against-one-s3vm	all-against-all-s3vm	1-step
4000	91,37 %	85,20 %	83,90 %	87,33 %	91,37 %
3000	90,57 %	83,18 %	81,12 %	86,22 %	90,96 %
2000	89,91 %	80,55 %	78,52 %	83,73 %	90,46 %
1000	88,43 %	75,67 %	71,61 %	78,76 %	88,55 %
500	86,23 %	72,01 %	66,96 %	74,95 %	85,55 %
200	82,11 %	64,81 %	58,60 %	68,02 %	79,81 %
100	72,54 %	59,60 %	52,19 %	62,64 %	69,19 %
50	62,06 %	51,92 %	47,28 %	56,51 %	58,80 %

Tabla 3.1: Resultados para BankSearch: Comparativa de técnicas S^3VM multiclase

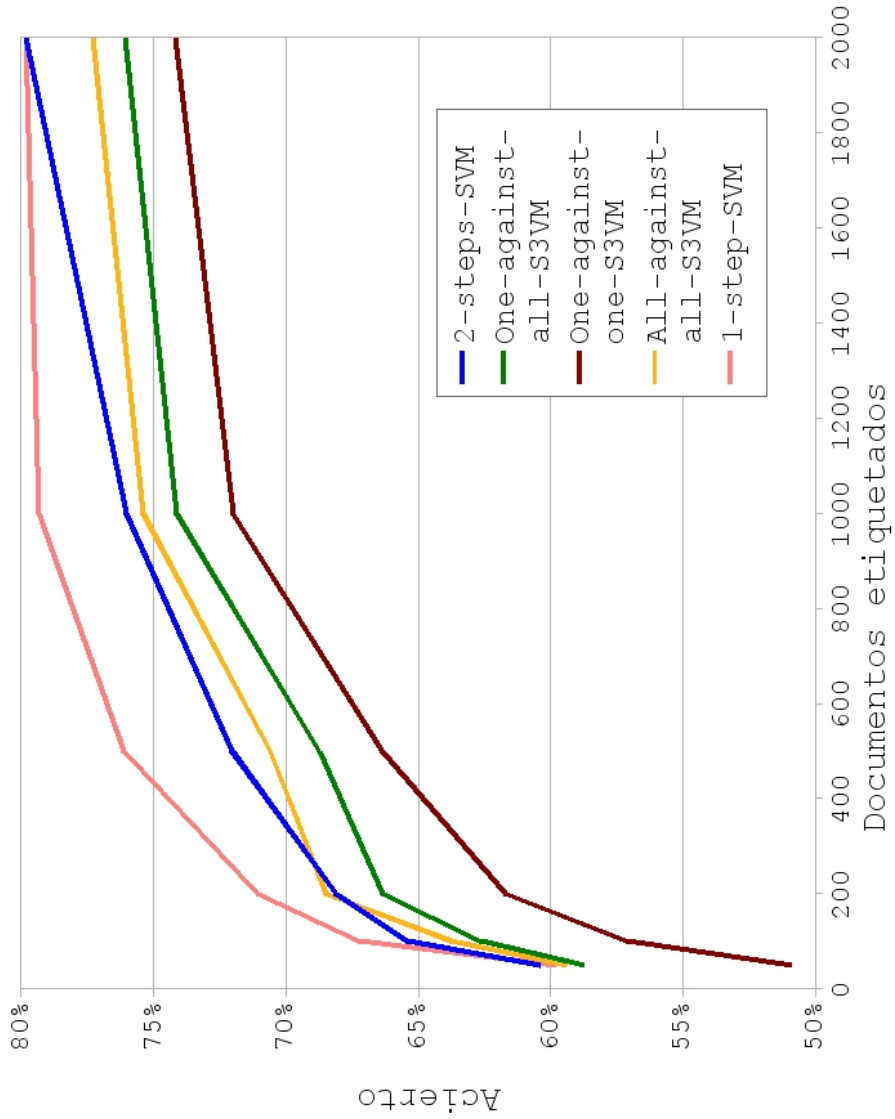

Figura 3.10: Resultados para WebKB: Comparativa de técnicas S^3VM multiclase

núm. etiquetados \ algoritmo	2-steps	one-against-all-s3vm	one-against-one-s3vm	all-against-all-s3vm	1-step
2000	79,83 %	76,05 %	74,19 %	77,28 %	79,83 %
1000	76,03 %	74,12 %	72,00 %	75,40 %	79,32 %
500	72,03 %	68,74 %	66,38 %	70,59 %	76,12 %
200	68,12 %	66,33 %	61,68 %	68,47 %	71,06 %
100	65,39 %	62,63 %	57,09 %	63,76 %	67,21 %
50	60,37 %	58,70 %	50,87 %	59,36 %	59,78 %

Tabla 3.2: Resultados para WebKB: Comparativa de técnicas S^3VM multiclase

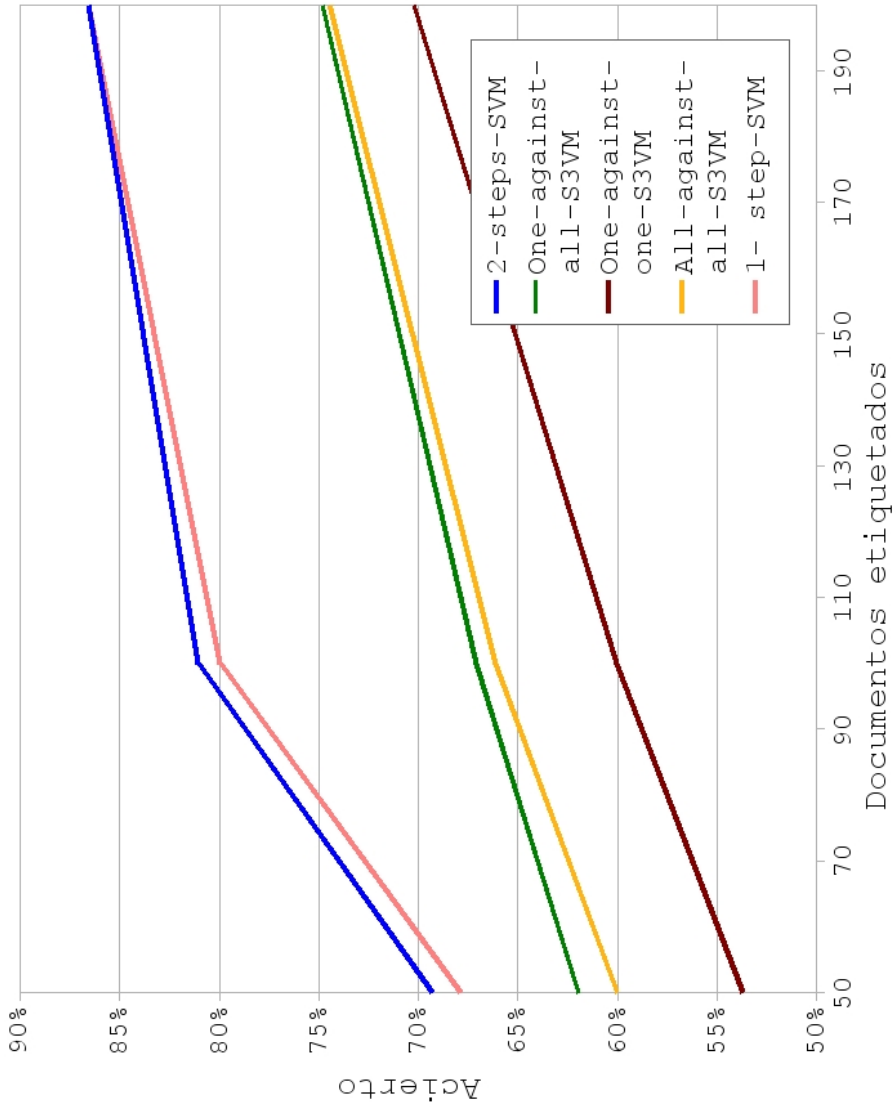

Figura 3.11: Resultados para Yahoo! Science: Comparativa de técnicas S^3VM multiclase

núm. etiquetados \ algoritmo	2-steps	one-against-all-s3vm	one-against-one-s3vm	all-against-all-s3vm	1-step
200	86,56 %	74,83 %	70,24 %	74,49 %	86,56 %
100	81,06 %	67,08 %	60,05 %	66,12 %	79,97 %
50	69,30 %	61,96 %	53,68 %	60,01 %	67,88 %

Tabla 3.3: Resultados para Yahoo! Science: Comparativa de técnicas S^3VM multiclase

Capítulo 4

Conclusiones y trabajo futuro

En este trabajo se ha realizado un estudio comparativo de clasificación semisupervisada multiclase de páginas web mediante SVM. Ante la problemática de la clasificación de páginas web, la cual se ha considerado una tarea semisupervisada y multiclase, se ha visto la necesidad de proponer técnicas que se adapten a este entorno, ya que SVM únicamente soluciona problemas supervisados y binarios por naturaleza, y apenas existen soluciones en la literatura.

En primer lugar, se han comparado los resultados ofrecidos por SVM con los de otras conocidas técnicas semisupervisadas. Los resultados muestran que la efectividad de SVM es superior a la del resto.

Con el objetivo de proponer una técnica apropiada para la trasformación de SVM a semisupervisado y multiclase, se han comparado y evaluado múltiples aproximaciones. Por una parte, se han introducido dos nuevas técnicas para S^3VM multiclase: *2-steps-SVM*, basado en SVM supervisado multiclase, y *all-against-all-S^3 VM*, que combina clasificadores SVM semisupervisados binarios. Por otra parte, se han aplicado por primera vez las técnicas *one-against-all* y *one-against-one* sobre clasificación semisupervisada, con sus variantes *one-against-all-S^3 VM* y *one-against-one-S^3 VM*, respectivamente, gracias a la combinación de clasificadores semisupervisados binarios.

Entre los métodos mencionados arriba, el denominado *2-steps-SVM*, basado en la aplicación de un SVM supervisado multiclase dos veces, muestra una clara superioridad respecto a las demás. Los peores resultados han sido para *one-against-one-S^3 VM*, debido a la existencia de ruido en su proceso de aprendizaje, por lo que no se considera adecuada su aplicación a entornos semisupervisados.

Por otro lado, la no inclusión de documentos no etiquetados en la fase de aprendizaje, aplicada mediante la técnica supervisada multiclase *1-step-*

SVM, ha mostrado que en algunas ocasiones puede afectar de forma positiva. Ignorar los documentos no etiquetados para aprender ha resultado mejor cuando las clases son más homogéneas. Para colecciones heterogéneas, sin embargo, los resultados de *2-steps-SVM* son ligeramente superiores.

Entre los algoritmos que combinan clasificadores binarios, *all-against-all-S^3VM* ha demostrado la mayor efectividad, aunque el gran número de clasificadores a considerar hace que su coste computacional aumente, por lo que su mejora en cuanto a eficiencia resultaría un interesante avance.

Como conclusión, cabe destacar la superioridad de los métodos que combinan clasificadores supervisados multiclase ante los que combinan clasificadores semisupervisados binarios ante este tipo de tareas. La gran efectividad obtenida mediante aquellos metodos hace que sea interesante su aplicación a los problemas de clasificación semisupervisada multiclase de páginas web. Además, los buenos resultados mostrados por la técnica *1-step-SVM* muestran que la inclusión de documentos no etiquetados en fase de aprendizaje para entornos semisupervisados puede no ser adecuada para problemas multiclase.

Como trabajo futuro, quedan por comparar los resultados respecto al algoritmo semisupervisado multiclase nativo introducido por Yajima & Kuo (2006).

Capítulo 5

Publicaciones del autor relacionadas con el trabajo

- Zubiaga, A., & Fresno, V. 2008. Comparativa de Aproximaciones a SVM Semisupervisado Multiclase para Clasificación de Páginas Web. *SEPLN, Sociedad Española para el Procesamiento del Lenguaje Natural*. Pendiente de publicación.
- Zubiaga, A., & Fresno, V. 2008. Semi-supervised Multiclass SVM for Web Page Classification: A Comparative Study. *Proceedings of ICDM 2008, IEEE International Conference on Data Mining*. Pendiente de aceptación.

Capítulo 6

Agradecimientos

Este trabajo ha sido subvencionado parcialmente por el proyecto QEAVis-Catiex (TIN2007-67581-C02-01) del Ministerio de Ciencia e Innovación, y por la Consejería de Educación de la Comunidad de Madrid.

Bibliografía

Amine, B. M., & Mimoun, M. 2007. Wordnet based cross-language text categorization. Proceedings of AICCSA '07, IEEE/ACS International Conference on Computer Systems and Applications.

Amitay, Einat, Carmel, David, Darlow, Adam, Lempel, Ronny, & Soffer, Aya. 2003. The connectivity sonar: detecting site functionality by structural patterns. *Pages 38–47 of: Hypertext '03: Proceedings of the fourteenth acm conference on hypertext and hypermedia.* New York, NY, USA: ACM.

Ando, Rie Kubota, & Zhang, Tong. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. mach. learn. res.*, **6**, 1817–1853.

Ando, Rie Kubota, & Zhang, Tong. 2007. Two-view feature generation model for semi-supervised learning. *Pages 25–32 of: Icml '07: Proceedings of the 24th international conference on machine learning.* New York, NY, USA: ACM.

Androutsopoulos, Ion, Koutsias, John, Chandrinos, Konstantinos V., & Spyropoulos, Constantine D. 2000. An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. *Pages 160–167 of: Sigir '00: Proceedings of the 23rd annual international acm sigir conference on research and development in information retrieval.* New York, NY, USA: ACM.

Angelova, Ralitsa, & Weikum, Gerhard. 2006. Graph-based text classification: learn from your neighbors. *Pages 485–492 of: Sigir '06: Proceedings of the 29th annual international acm sigir conference on research and development in information retrieval.* New York, NY, USA: ACM.

Apté, Chidanand, Damerau, Fred, & Weiss, Sholom M. 1994. Automated learning of decision rules for text categorization. *Acm trans. inf. syst.*, **12**(3), 233–251.

Attardi, Giuseppe, Gulli, Antonio, & Sebastiani, Fabrizio. 1999. Automatic web page categorization by link and context analysis. *Pages 105–119 of:* Hutchison, Chris, & Lanzarone, Gaetano (eds), *Thai-99*.

Aue, A., & Gamon, M. 2005. Customizing sentiment classifiers to a new domain. a case study. Proceedings of RANLP-05, the International Conference on Recent Advances in Natural Language Processing.

Baker, L. Douglas, & McCallum, Andrew Kachites. 1998. Distributional clustering of words for text classification. *Pages 96–103 of: Sigir '98: Proceedings of the 21st annual international acm sigir conference on research and development in information retrieval*. New York, NY, USA: ACM.

Bauer, Frank, Pereverzev, Sergei, & Rosasco, Lorenzo. 2007. On regularization algorithms in learning theory. *J. complex.*, **23**(1), 52–72.

Belkin, Mikhail, Niyogi, Partha, & Sindhwani, Vikas. 2006. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. mach. learn. res.*, **7**, 2399–2434.

Belkin, Nicholas J., & Croft, W. Bruce. 1992. Information filtering and information retrieval: two sides of the same coin? *Communications of the acm*, **35**(12), 29–38.

Bie, T. De, & Cristianini, N. 2004. *Convex methods for transduction*.

Bie, T. De, & Cristianini, N. 2006. Semi-supervised learning using semi-definite programming. *Pages 119–136 of: Semi-supervised learning*. Cambridge, MA: MIT Press.

Billsus, Daniel, & Pazzani, Michael J. 2000. User modeling for adaptive news access. *User modeling and user-adapted interaction*, **10**(2-3), 147–180.

Blitzer, John, Dredze, Mark, & Pereira, Fernando. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. *Pages 440–447 of: Proceedings of the 45th annual meeting of the association of computational linguistics*. Prague, Czech Republic: Association for Computational Linguistics.

Blum, Avrim, & Chawla, Shuchi. 2001. Learning from labeled and unlabeled data using graph mincuts. *Pages 19–26 of: Icml '01: Proceedings of the eighteenth international conference on machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Blum, Avrim, & Mitchell, Tom. 1998. Combining labeled and unlabeled data with co-training. *Pages 92–100 of: Colt' 98: Proceedings of the eleventh annual conference on computational learning theory.* New York, NY, USA: ACM.

Blum, Avrim, Lafferty, John, Rwebangira, Mugizi Robert, & Reddy, Rajashekar. 2004. Semi-supervised learning using randomized mincuts. *Page 13 of: Icml '04: Proceedings of the twenty-first international conference on machine learning.* New York, NY, USA: ACM.

Bolelli, Levent, Ertekin, Seyda, Zhou, Ding, & Giles, C. Lee. 2007. A clustering method for web data with multi-type interrelated components. *Pages 1121–1122 of: Www '07: Proceedings of the 16th international conference on world wide web.* New York, NY, USA: ACM.

Bordes, Antoine, Bottou, Léon, Gallinari, Patrick, & Weston, Jason. 2007. Solving multiclass support vector machines with larank. *Pages 89–96 of: Icml '07: Proceedings of the 24th international conference on machine learning.* New York, NY, USA: ACM.

Borman, Sean. 2004 (July). *The expectation maximization algorithm – a short tutorial.* Introduces the Expectation Maximization (EM) algorithm and fleshes out the basic mathematical results, inclusing a proof of convergence. The Generalized EM algorithm is also introduced.

Boser, Bernhard E., Guyon, Isabelle M., & Vapnik, VladimirÑ. 1992. A training algorithm for optimal margin classifiers. *Pages 144–152 of: Colt '92: Proceedings of the fifth annual workshop on computational learning theory.* New York, NY, USA: ACM.

Brefeld, Ulf, & Scheffer, Tobias. 2006. Semi-supervised learning for structured output variables. *Pages 145–152 of: Icml '06: Proceedings of the 23rd international conference on machine learning.* New York, NY, USA: ACM.

Buckley, Chris, Salton, Gerard, & Allan, James. 1994. The effect of adding relevance information in a relevance feedback environment. *Pages 292–300 of: Sigir '94: Proceedings of the 17th annual international acm sigir conference on research and development in information retrieval.* New York, NY, USA: Springer-Verlag New York, Inc.

Calado, Pável, Cristo, Marco, Moura, Edleno, Ziviani, Nivio, Ribeiro-Neto, Berthier, & Gonçalves, Marcos André. 2003. Combining link-based and content-based methods for web document classification. *Pages 394–401 of: Cikm '03: Proceedings of the twelfth international conference on information and knowledge management.* New York, NY, USA: ACM.

Campbell, C. 2000. *Kernel methods: a survey of current techniques.*

Caropreso, Maria Fernanda, Matwin, Stan, & Sebastiani, Fabrizio. 2001. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. 78–102.

Carpuat, Marine, Su, Weifeng, & Wu, Dekai. 2004. Augmenting ensemble classification for word sense disambiguation with a kernel pca model. *Pages 88–92 of:* Mihalcea, Rada, & Edmonds, Phil (eds), *Senseval-3: Third international workshop on the evaluation of systems for the semantic analysis of text.* Barcelona, Spain: Association for Computational Linguistics.

Cavnar, William B., & Trenkle, John M. 1994. N-gram-based text categorization. *Pages 161–175 of: Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval.*

Ceci, Michelangelo, & Malerba, Donato. 2007. Classifying web documents in a hierarchy of categories: a comprehensive study. *J. intell. inf. syst.*, **28**(1), 37–78.

Chakrabarti, Soumen, Dom, Byron, & Indyk, Piotr. 1998. Enhanced hypertext categorization using hyperlinks. *Sigmod rec.*, **27**(2), 307–318.

Chapelle, O., & Zien, A. 2005. *Semi-supervised classification by low density separation.*

Chapelle, O., Weston, J., & Schölkopf, B. 2003. Cluster kernels for semi-supervised learning. NIPS, vol. 15.

Chapelle, Olivier, Sindhwani, Vikas, & Keerthi, S. Sathiya. 2006a. Branch and bound for semi-supervised support vector machines. *Pages 217–224 of: Nips.*

Chapelle, Olivier, Chi, Mingmin, & Zien, Alexander. 2006b. A continuation method for semi-supervised svms. *Pages 185–192 of: Icml '06: Proceedings of the 23rd international conference on machine learning.* New York, NY, USA: ACM.

Chekuri, C., Goldwasser, M., Raghavan, Prabhakar, & Upfal, E. 1996. Web search using automatic classification. *In: Proceedings of WWW-96, 6th international conference on the world wide web.*

Chen, Chih-Ming, Lee, Hahn-Ming, & Hwang, Cheng-Wei. 2005. A hierarchical neural network document classifier with linguistic feature selection. *Applied intelligence*, **23**(3), 277–294.

Clack, Chris, Farringdon, Johnny, Lidwell, Peter, & Yu, Tina. 1997. Autonomous document classification for business. *Pages 201–208 of: Agents '97: Proceedings of the first international conference on autonomous agents*. New York, NY, USA: ACM.

Cohen, William W. 1995. Learning to classify english text with ilp methods.

Cohen, William W. 2002. *Improving a page classifier with anchor extraction and link analysis*.

Cohen, William W., & Hirsh, Haym. 1998. Joins that generalize: text classification using Whirl. *Pages 169–173 of:* Agrawal, Rakesh, Stolorz, Paul E., & Piatetsky-Shapiro, Gregory (eds), *Proceedings of KDD-98, 4th international conference on knowledge discovery and data mining*. New York, US: AAAI Press, Menlo Park, US.

Cohen, William W., & Singer, Yoram. 1999. Context-sensitive learning methods for text categorization. *Acm trans. inf. syst.*, **17**(2), 141–173.

Collins, M., & Singer, Y. 1999. *Unsupervised models for named entity classification*.

Collobert, Ronan, Sinz, Fabian, Weston, Jason, & Bottou, Léon. 2006. Trading convexity for scalability. *Pages 201–208 of: Icml '06: Proceedings of the 23rd international conference on machine learning*. New York, NY, USA: ACM.

Corduneanu, Adrian, & Jaakkola, Tommi. 2001. Stable mixing of complete and incomplete information. NDLTD Union Catalog.

Cortes, Corinna, & Vapnik, Vladimir. 1995. Support-vector networks. *Machine learning*, **20**(3), 273–297.

Cui, Hang, Mittal, Vibhu O., & Datar, Mayur. 2006. Comparative experiments on sentiment classification for online product reviews. *In: Aaai*.

Dagan, Ido, Karov, Yael, & Roth, Dan. 1997. Mistake-driven learning in text categorization. *Pages 55–63 of:* Cardie, Claire, & Weischedel, Ralph (eds), *Proceedings of EMNLP-97, 2nd conference on empirical methods in natural language processing*. Providence, US: Association for Computational Linguistics, Morristown, US.

Dara, R., Kremer, S., & Stacey, D. 2002. Clustering unlabeled data with soms improves classification of labeled real-world data. Proceedings of the World Congress on Computational Intelligence.

DE LOUPY, C., EL-BÏ¿½ZE, M., & MARTEAU, P.-F. 2000. *Using semantic classification trees for wsd.*

DE MELO, GERARD, & SIERSDORFER, STEFAN. 2007. Multilingual text classification using ontologies. *Pages 541–548 of:* AMATI, GIAMBATTISTA, CARPINETO, CLAUDIO, & ROMANO, GIOVANNI (eds), *Advances in information retrieval, 29th european conference on ir research, ecir 2007, rome, italy, april 2-5, 2007, proceedings.* Lecture Notes in Computer Science, vol. 4425. Springer.

DE SA, VIRGINIA R. 1993. Learning classification with unlabeled data. *Pages 112–119 of:* COWAN, JACK D., TESAURO, GERALD, & ALSPECTOR, JOSHUA (eds), *Proc. nips'93, neural information processing systems.* San Francisco, CA: Morgan Kaufmann Publishers.

DELALLEAU, O., BENGIO, Y., & ROUX, N. L. 2005. Efficient non-parametric function induction in semi-supervised learning. Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics.

DEMIRIZ, A., & BENNETT, K. 2000. *Optimization approaches to semisupervised learning.*

DEMIRIZ, A., BENNETT, K., & EMBRECHTS, M. 1999. *Semi-supervised clustering using genetic algorithms.*

DEMPSTER, A. P., LAIRD, N. M., & RUBIN, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. series b (methodological)*, **39**(1), 1–38.

DÖRFLER, A., EILERT, S., MENTRUP, A., MÜLLER, M. E., ROLF, R., ROLLINGER, C.-R., SIEVERTSEN, F., & TRENKAMP, F. 2001. Bikini: User adaptive news classification in the world wide web. Proceedings of Workshop Machine Learning for User Modeling, 8th International Conference on User Modeling.

DRUCKER, H., VAPNIK, V., & WU, D. 1999. Automatic text categorization and its applications to text retrieval. IEEE Trans. Neural Networks.

DUMAIS, SUSAN, & CHEN, HAO. 2000. Hierarchical classification of web content. *Pages 256–263 of: Sigir '00: Proceedings of the 23rd annual international acm sigir conference on research and development in information retrieval.* New York, NY, USA: ACM.

DUMAIS, SUSAN, PLATT, JOHN, HECKERMAN, DAVID, & SAHAMI, MEHRAN. 1998. Inductive learning algorithms and representations for text categorization. *Pages 148–155 of: Cikm '98: Proceedings of the seventh international conference on information and knowledge management.* New York, NY, USA: ACM.

Escudero, Gerard, Màrquez, Lluís, & Rigau, German. 2000. Boosting applied toe word sense disambiguation. *Pages 129–141 of: Ecml '00: Proceedings of the 11th european conference on machine learning.* London, UK: Springer-Verlag.

Ester, Martin, Kriegel, Hans-Peter, & Schubert, Matthias. 2002. Web site mining: a new way to spot competitors, customers and suppliers in the world wide web. *Pages 249–258 of: Kdd '02: Proceedings of the eighth acm sigkdd international conference on knowledge discovery and data mining.* New York, NY, USA: ACM.

Farmer, Michael E., & Jain, Anil K. 2004. A wrapper-based approach to image segmentation and classification. *Pages 106–109 of: Icpr '04: Proceedings of the pattern recognition, 17th international conference on (icpr'04) volume 2.* Washington, DC, USA: IEEE Computer Society.

Fisher, Michelle, & Everson, Richard. 2003. When are links useful? experiments in text classification. Proceedings of Advances in Information Retrieval: 25th European Conference on IR Research.

Fowlkes, Charless, Belongie, Serge, Chung, Fan, & Malik, Jitendra. 2004. Spectral grouping using the nyström method. *Ieee trans. pattern anal. mach. intell.*, **26**(2), 214–225.

Fuhr, N., Hartmann, S., Knorz, G., Lustig, G., Schwantner, M., & Tzeras, K. 1991. Air/x, a rule-based automated indexing system for large subject fields. Proceedings of RIAO-91, 3rd International Conference Recherche d'Information Assistee par Ordinateur.

Fuhr, Norbert, & Buckley, Chris. 1991. A probabilistic learning approach for document indexing. *Acm trans. inf. syst.*, **9**(3), 223–248.

Fujino, Akinori, Ueda, Naonori, & Saito, Kazumi. 2008. Semisupervised learning for a hybrid generative/discriminative classifier based on the maximum entropy principle. *Ieee trans. pattern anal. mach. intell.*, **30**(3), 424–437.

Fürnkranz, Johannes. 1999. Exploiting structural information for text classification on the www. *Pages 487–498 of: Ida '99: Proceedings of the third international symposium on advances in intelligent data analysis.* London, UK: Springer-Verlag.

Fürnkranz, Johannes. 2001. Hyperlink ensembles: A case study in hypertext classification. *Journal of information fusion.*

Galavotti, Luigi, Sebastiani, Fabrizio, & Simi, Maria. 2000. Experiments on the use of feature selection and negative evidence in automated

text categorization. *Pages 59–68 of: Ecdl '00: Proceedings of the 4th european conference on research and advanced technology for digital libraries.* London, UK: Springer-Verlag.

Gale, W. A., Church, K. W., & Yarowsky, D. 1993. A method for disambiguating word senses in a large corpus. *Computers and the humanities.*

García Adeva, J. J., de Ipiña, Diego López, & Calvo, R. 2005. Multilingual approaches to text categorisation. *The european journal for the informatics professional*, **VI**(3), 43 – 51.

Getz, G., Shental, N., & Domany, E. 2005. Semi-supervised learning - a statistical physics approach. Proceedings of 22nd ICML Workshop on Learning with Partially Classified Training Data.

Ghani, Rayid, Slattery, Seán, & Yang, Yiming. 2001. Hypertext categorization using hyperlink patterns and meta data. *Pages 178–185 of: Icml '01: Proceedings of the eighteenth international conference on machine learning.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Gliozzo, A., & Strapparava, C. Cross language text categorization by acquiring multilingual domain models from comparable corpora. Proceedings of the ACL 2005 Workshop: Building and Using Parallel Corpora: Data-driven Machine Translation and Beyond.

Gliozzo, Alfio, & Strapparava, Carlo. 2006. Exploiting comparable corpora and bilingual dictionaries for cross-language text categorization. *Pages 553–560 of: Acl '06: Proceedings of the 21st international conference on computational linguistics and the 44th annual meeting of the acl.* Morristown, NJ, USA: Association for Computational Linguistics.

Glover, Eric J., Tsioutsiouliklis, Kostas, Lawrence, Steve, Pennock, David M., & Flake, Gary W. 2002. Using web structure for classifying and describing web pages. *Pages 562–569 of: Www '02: Proceedings of the 11th international conference on world wide web.* New York, NY, USA: ACM.

Goldberg, A., & Zhu, X. 2004. Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. HLT-NAACL 2006 Workshop on Textgraphs: Graph- based Algorithms for Natural Language Processing.

Goldberg, A., Zhu, X., & Wright, S. 2007. Dissimilarity in graph-based semi-supervised classification. Proceedings of AISTATS, 11th International Conference on Artificial Intelligence and Statistics.

Goldman, Sally A., & Zhou, Yan. 2000. Enhancing supervised learning with unlabeled data. *Pages 327–334 of: Icml '00: Proceedings of the seventeenth international conference on machine learning.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Golub, K., & Ardō, A. 2005. Importance of html structural elements and metadata in automated subject classification. Proceedings of the 9th European Conference on Research and Advanced Technology for Digital Libraries.

Gomez, Jose M., Buenaga, Manuel De, & Giráldez, Ignacio. 2004. Text categorization for internet content filtering. *Inteligencia artificial: Revista iberoamericana de inteligencia artificial.*

Harabagiu, S., Pasca, M., & Maiorano, S. 2000. *Experiments with open-domain textual question answering.*

Haveliwala, Taher H. 2002. Topic-sensitive pagerank. *Pages 517–526 of: Www '02: Proceedings of the 11th international conference on world wide web.* New York, NY, USA: ACM.

Hermjakob, Ulf. 2001. Parsing and question classification for question answering. *Pages 1–6 of: Proceedings of the workshop on arabic language processing.* Morristown, NJ, USA: Association for Computational Linguistics.

Heymann, P., Ramage, D., & Garcia-Molina, H. 2008. Social tag prediction. The 31st Annual International ACM SIGIR Conference.

Hsu, C., & Lin, C. 2001. *A comparison of methods for multi-class support vector machines.*

Hull, David. 1994. Improving text retrieval for the routing problem using latent semantic indexing. *Pages 282–291 of: Sigir '94: Proceedings of the 17th annual international acm sigir conference on research and development in information retrieval.* New York, NY, USA: Springer-Verlag New York, Inc.

Joachims, Thorsten. 1997. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. *Pages 143–151 of:* Fisher, Douglas H. (ed), *Proceedings of ICML-97, 14th international conference on machine learning.* Nashville, US: Morgan Kaufmann Publishers, San Francisco, US.

Joachims, Thorsten. 1998. Text categorization with support vector machines: learning with many relevant features. *Pages 137–142 of:* Nédellec, Claire, & Rouveirol, Céline (eds), *Proceedings of ECML-98,*

10th european conference on machine learning. Chemnitz, DE: Springer Verlag, Heidelberg, DE.

Joachims, Thorsten. 1999. Transductive inference for text classification using support vector machines. *Pages 200–209 of: Icml '99: Proceedings of the sixteenth international conference on machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Jones, Rosie. 2005. *Learning to extract entities from labeled and unlabeled text*.

Käki, Mika. 2005. Optimizing the number of search result categories. *Pages 1517–1520 of: Chi '05: Chi '05 extended abstracts on human factors in computing systems*. New York, NY, USA: ACM.

Kan, Min-Yen. 2004. Web page classification without the web page. *Pages 262–263 of: Www alt. '04: Proceedings of the 13th international world wide web conference on alternate track papers & posters*. New York, NY, USA: ACM.

Kan, Min-Yen, & Thi, Hoang Oanh Ñguyen. 2005. Fast webpage classification using url features. *Pages 325–326 of: Cikm '05: Proceedings of the 14th acm international conference on information and knowledge management*. New York, NY, USA: ACM.

Karamon, Jun, Matsuo, Yutaka, Yamamoto, Hikaru, & Ishizuka, Mitsuru. 2007. Generating social network features for link-based classification. PKDD.

Kennedy, Alistair, & Inkpen, Diana. 2005. Sentiment classification of movie and product reviews using contextual valence shifters. Proceedings of FINEXIN 2005, Workshop on the Analysis of Informal and Formal Information Exchange during Negotiations.

Kennedy, Alistair, & Inkpen, Diana. 2006. Sentiment classification of movie reviews using contextual valence shifters. *Computational intelligence*.

Kleinberg, Jon M. 1999. Authoritative sources in a hyperlinked environment. *J. acm*, **46**(5), 604–632.

Kohavi, R., & John, G. 1998. The wrapper approach. *Pages 33–50 of:* Liu, H., & Motoda, H. (eds), *Feature selection for knowledge discovery and data mining*.

Kolcz, Aleksander, & Chowdhury, Abdur. 2007. Avoidance of model re-induction in svm-based feature selection for text categorization. *Pages 889–894 of:* Veloso, Manuela M. (ed), *Ijcai*.

Kovacevic, Milos, Diligenti, Michelangelo, Gori, Marco, & Milutinovic, Veljko. 2004. Visual adjacency multigraphs - a novel approach for a web page classification. Proceedings of SAWM04 workshop.

Krishnapuram, B., Williams, D., Xue, Y., Hartemink, A., Carin, L., & Figueiredo, M. 2005. On semi-supervised classification. *Advances in neural information processing systems.*

Kules, Bill, Kustanowitz, Jack, & Shneiderman, Ben. 2006. Categorizing web search results into meaningful and stable categories using fast-feature techniques. *Pages 210–219 of: Jcdl '06: Proceedings of the 6th acm/ieee-cs joint conference on digital libraries.* New York, NY, USA: ACM.

Kwok, Cody C. T., Etzioni, Oren, & Weld, Daniel S. 2001. Scaling question answering to the web. *Pages 150–161 of: Www '01: Proceedings of the 10th international conference on world wide web.* New York, NY, USA: ACM.

Kwon, Oh-Woog, & Lee, Jong-Hyeok. 2000. Web page classification based on k-nearest neighbor approach. *Pages 9–15 of: Iral '00: Proceedings of the fifth international workshop on on information retrieval with asian languages.* New York, NY, USA: ACM.

Kwon, Oh-Woog, & Lee, Jong-Hyeok. 2003. Text categorization based on k-nearest neighbor approach for web site classification. *Inf. process. manage.*, **39**(1), 25–44.

Lam, Savio L. Y., & Lee, Dik Lun. 1999. Feature reduction for neural network based text categorization. *Pages 195–202 of: Dasfaa '99: Proceedings of the sixth international conference on database systems for advanced applications.* Washington, DC, USA: IEEE Computer Society.

Lam, W., Low, K. F., & Ho, C. Y. 1997. Using a bayesian network induction approach for text categorization. Proceedings of IJCAI-97, 15th International Joint Conference on Artificial Intelligence.

Lam, Wai, & Ho, Chao Yang. 1998. Using a generalized instance set for automatic text categorization. *Pages 81–89 of: Sigir '98: Proceedings of the 21st annual international acm sigir conference on research and development in information retrieval.* New York, NY, USA: ACM.

Larkey, Leah S., & Croft, W. Bruce. 1996. Combining classifiers in text categorization. *Pages 289–297 of: Sigir '96: Proceedings of the 19th annual international acm sigir conference on research and development in information retrieval.* New York, NY, USA: ACM.

Lee, Chung-Hong, & Yang, Hsin-Chang. 2007. Implementation of unsupervised and supervised learning systems for multilingual text categorization. *Pages 377–382 of: Itng '07: Proceedings of the international conference on information technology.* Washington, DC, USA: IEEE Computer Society.

Lee, Chung-Hong, Yang, Hsin-Chang, Chen, Ting-Chung, & Ma, Sheng-Min. 2006. A comparative study on supervised and unsupervised learning approaches for multilingual text categorization. *Pages 511–514 of: Icicic '06: Proceedings of the first international conference on innovative computing, information and control.* Washington, DC, USA: IEEE Computer Society.

Lewis, David D. 1992. An evaluation of phrasal and clustered representations on a text categorization task. *Pages 37–50 of: Sigir '92: Proceedings of the 15th annual international acm sigir conference on research and development in information retrieval.* New York, NY, USA: ACM.

Lewis, David D., & Catlett, Jason. 1994. Heterogeneous uncertainty sampling for supervised learning. *Pages 148–156 of:* Cohen, William W., & Hirsh, Haym (eds), *Proceedings of ICML-94, 11th international conference on machine learning.* New Brunswick, US: Morgan Kaufmann Publishers, San Francisco, US.

Lewis, David D., & Ringuette, Marc. 1994. A comparison of two learning algorithms for text categorization. *Pages 81–93 of: Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval.*

Li, Hang, & Yamanishi, Kenji. 2002. Text classification using esc-based stochastic decision lists. *Inf. process. manage.*, **38**(3), 343–361.

Li, Yonghong, & Jain, A.K. 1998. Classification of text documents. *Pattern recognition, 1998. proceedings. fourteenth international conference on*, **2**(Aug), 1295–1297 vol.2.

Liddy, Elizabeth D., Paik, Woojin, & Yu, Edmund S. 1994. Text categorization for multiple users based on semantic features from a machine-readable dictionary. *Acm trans. inf. syst.*, **12**(3), 278–295.

Lindemann, Christoph, & Littig, Lars. 2006. Coarse-grained classification of web sites by their structural properties. *Pages 35–42 of: Widm '06: Proceedings of the 8th annual acm international workshop on web information and data management.* New York, NY, USA: ACM.

Lu, Qing, & Getoor, Lise. 2003. Link-based classification using labeled and unlabeled data. Proceedings of ICML Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining.

Maeireizo, Beatriz, Litman, Diane, & Hwa, Rebecca. 2004. Co-training for predicting emotions with spoken dialogue data. *Page 28 of: Proceedings of the acl 2004 on interactive poster and demonstration sessions*. Morristown, NJ, USA: Association for Computational Linguistics.

Manning, Christopher D., & Schütze, Hinrich. 1999. *Foundations of statistical natural language processing*. Cambridge, Massachusetts: The MIT Press.

Maria, Nuno, & Silva, Mário J. 2001. Theme-based retrieval of web news. *Pages 26–37 of: Selected papers from the third international workshop webdb 2000 on the world wide web and databases*. London, UK: Springer-Verlag.

Masand, Brij. 1994. Optimizing confidence of text classification by evolution of symbolic expressions. 445–458.

Mejía-Lavalle, M., Morales, E.F., & Rodriguez, G. 2006. Fast feature selection method for continuous attributes with nominal class. *Artificial intelligence, 2006. micai '06. fifth mexican international conference on*, Nov., 142–150.

Mitchell, Tom M. 1997. *Machine learning*. New York: McGraw-Hill.

Mittermayer, Marc-Andre, & Knolmayer, Gerhard F. 2006. Newscats: A news categorization and trading system. *icdm*, **0**, 1002–1007.

Mladenić, Dunja. 1998. Feature subset selection in text-learning. *Pages 95–100 of: Ecml '98: Proceedings of the 10th european conference on machine learning*. London, UK: Springer-Verlag.

Moulinier, Isabelle, & Ganascia, Jean-Gabriel. 1996. Applying an existing machine learning algorithm to text categorization. *Pages 343–354 of: Connectionist, statistical, and symbolic approaches to learning for natural language processing*. London, UK: Springer-Verlag.

Moulinier, Isabelle, Raskinis, Gailius, & Ganascia, Jean-Gabriel. 1996. Text categorization: a symbolic approach. Proceedings of SDAIR-96, 5th Annual Symposium on Document Analysis and Information Retrieval.

Nather, Peter. 2005. *N-gram based text categorization*. M.Phil. thesis, Univerzita Komenského - Fakulta matematiky, Univerzita Komenského v Bratislave.

Ng, Hwee Tou, Goh, Wei Boon, & Low, Kok Leong. 1997. Feature selection, perceptron learning, and a usability case study for text categorization. *Pages 67–73 of: Sigir '97: Proceedings of the 20th annual international acm sigir conference on research and development in information retrieval*. New York, NY, USA: ACM.

Nie, Lan, Davison, Brian D., & Qi, Xiaoguang. 2006. Topical link analysis for web search. *Pages 91–98 of: Sigir '06: Proceedings of the 29th annual international acm sigir conference on research and development in information retrieval*. New York, NY, USA: ACM.

Nigam, Kamal, & Ghani, Rayid. 2000. Analyzing the effectiveness and applicability of co-training. *Pages 86–93 of: Cikm '00: Proceedings of the ninth international conference on information and knowledge management*. New York, NY, USA: ACM.

Nigam, Kamal, McCallum, Andrew Kachites, Thrun, Sebastian, & Mitchell, Tom. 2000. Text classification from labeled and unlabeled documents using em. *Mach. learn.*, **39**(2-3), 103–134.

Oh, Hyo-Jung, Myaeng, Sung Hyon, & Lee, Mann-Ho. 2000. A practical hypertext categorization method using links and incrementally available class information. *Pages 264–271 of: Sigir '00: Proceedings of the 23rd annual international acm sigir conference on research and development in information retrieval*. New York, NY, USA: ACM.

Page, L., Brin, S., Motwani, R., & Winograd, T. 1998. *The pagerank citation ranking: Bringing order to the web.*

Pang, Bo, & Lee, Lillian. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. *Page 271 of: Acl '04: Proceedings of the 42nd annual meeting on association for computational linguistics*. Morristown, NJ, USA: Association for Computational Linguistics.

Pang, Bo, & Lee, Lillian. 2005. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. *Pages 115–124 of: Acl '05: Proceedings of the 43rd annual meeting on association for computational linguistics*. Morristown, NJ, USA: Association for Computational Linguistics.

Pang, Bo, Lee, Lillian, & Vaithyanathan, Shivakumar. 2002. Thumbs up?: sentiment classification using machine learning techniques. *Pages 79–86 of: Emnlp '02: Proceedings of the acl-02 conference on empirical methods in natural language processing*. Morristown, NJ, USA: Association for Computational Linguistics.

Platt, J., Cristianini, N., & Shawe-Taylor, J. 2000. Large margin dags for multiclass classification. *Pages 547–553 of:* Solla, S.A., Leen, T.K., & Mueller, K.-R. (eds), *Advances in neural information processing systems 12.*

Prescher, Detlef. 2005 (March). *A tutorial on the expectation-maximization algorithm including maximum-likelihood estimation and em training of probabilistic context-free grammars.*

Qi, Xiaoguang, & Davison, Brian D. 2006. Knowing a web page by the company it keeps. *Pages 228–237 of: Cikm '06: Proceedings of the 15th acm international conference on information and knowledge management.* New York, NY, USA: ACM.

Qi, Xiaoguang, & Davison, Brian D. 2007. *Web page classification: Features and algorithms.* Technical Report LU-CSE-07-010. Lehigh University, Bethlehem, Pennsylvania, USA.

Ratsaby, Joel, & Venkatesh, Santosh S. 1995. Learning from a mixture of labeled and unlabeled examples with parametric side information. *Pages 412–417 of: Colt '95: Proceedings of the eighth annual conference on computational learning theory.* New York, NY, USA: ACM.

Rigutini, Leonardo, Maggini, Marco, & Liu, Bing. 2005. An em based training algorithm for cross-language text categorization. *Pages 529–535 of: Wi '05: Proceedings of the 2005 ieee/wic/acm international conference on web intelligence.* Washington, DC, USA: IEEE Computer Society.

Riloff, Ellen, Wiebe, Janyce, & Wilson, Theresa. 2003. Learning subjective nouns using extraction pattern bootstrapping. *Pages 25–32 of: Proceedings of the seventh conference on natural language learning at hlt-naacl 2003.* Morristown, NJ, USA: Association for Computational Linguistics.

Ruch, Patrick. 2004. Query translation by text categorization. *Page 686 of: Coling '04: Proceedings of the 20th international conference on computational linguistics.* Morristown, NJ, USA: Association for Computational Linguistics.

Ruiz, Miguel E., & Srinivasan, Padmini. 2002. Hierarchical text categorization using neural networks. *Inf. retr.*, **5**(1), 87–118.

Sable, Carl L., & Hatzivassiloglou, Vassileios. 2000. Text-based approaches for non-topical image categorization. *International journal on digital libraries.*

Schapire, Robert E., & Singer, Yoram. 2000. Boostexter: A boosting-based systemfor text categorization. *Machine learning*, **39**(2-3), 135–168.

Schapire, Robert E., Singer, Yoram, & Singhal, Amit. 1998. Boosting and rocchio applied to text filtering. *Pages 215–223 of: Sigir '98:*

Proceedings of the 21st annual international acm sigir conference on research and development in information retrieval. New York, NY, USA: ACM.

Schütze, Hinrich, Hull, David A., & Pedersen, Jan O. 1995. A comparison of classifiers and document representations for the routing problem. *Pages 229–237 of: Sigir '95: Proceedings of the 18th annual international acm sigir conference on research and development in information retrieval.* New York, NY, USA: ACM.

Sebastiani, Fabrizio. 2002. Machine learning in automated text categorization. *Acm comput. surv.*, **34**(1), 1–47.

Selamat, Ali, & Omatu, Sigeru. 2004. Web page feature selection and classification using neural networks. *Inf. sci. inf. comput. sci.*, **158**(1), 69–88.

Sindhwani, Vikas, Niyogi, Partha, & Belkin, Mikhail. 2005a. Beyond the point cloud: from transductive to semi-supervised learning. *Pages 824–831 of: Icml '05: Proceedings of the 22nd international conference on machine learning.* New York, NY, USA: ACM.

Sindhwani, Vikas, Niyogi, Partha, Belkin, Mikhail, & Keerthi, S. Sathiya. 2005b. Linear manifold regularization for large scale semi-supervised learning. Proceedings of the 22nd ICML Workshop on Learning with Partially Classified Training Data.

Sindhwani, Vikas, Keerthi, S. Sathiya, & Chapelle, Olivier. 2006. Deterministic annealing for semi-supervised kernel machines. *Pages 841–848 of: Icml '06: Proceedings of the 23rd international conference on machine learning.* New York, NY, USA: ACM.

Sinka, Mark, & Corne, David. 2002. *A large benchmark dataset for web document clustering.* Pages 881–890.

Slonim, N., & Tishby, N. 2001. The power of word clusters for text classification. *In: 23rd european colloquium on information retrieval research.*

Sun, Aixin, Lim, Ee-Peng, & Ng, Wee-Keong. 2002. Web classification using support vector machine. *Pages 96–99 of: Widm '02: Proceedings of the 4th international workshop on web information and data management.* New York, NY, USA: ACM.

Sun, Bingjun, Tan, Qingzhao, Mitra, Prasenjit, & Giles, C. Lee. 2007. Extraction and search of chemical formulae in text documents on the web. *Pages 251–260 of: Www '07: Proceedings of the 16th international conference on world wide web.* New York, NY, USA: ACM.

Tan, Chade-Meng, Wang, Yuan-Fang, & Lee, Chan-Do. 2002. The use of bigrams to enhance text categorization. *Inf. process. manage.*, **38**(4), 529–546.

Turney, Peter D. 2001. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. *Pages 417–424 of: Acl '02: Proceedings of the 40th annual meeting on association for computational linguistics*. Morristown, NJ, USA: Association for Computational Linguistics.

Tzeras, Kostas, & Hartmann, Stephan. 1993. Automatic indexing based on bayesian inference networks. *Pages 22–35 of: Sigir '93: Proceedings of the 16th annual international acm sigir conference on research and development in information retrieval*. New York, NY, USA: ACM.

Wang, Chong, & Wang, Wenyuan. 2005. Using term clustering and supervised term affinity construction to boost text classification. Proceedings of PAKDD.

Wang, Gang, Yeung, Dit-Yan, & Lochovsky, Frederick H. 2007a. A kernel path algorithm for support vector machines. *Pages 951–958 of: Icml '07: Proceedings of the 24th international conference on machine learning*. New York, NY, USA: ACM.

Wang, Li, Zhu, Ji, & Zou, Hui. 2007b. Hybrid huberized support vector machines for microarray classification. *Pages 983–990 of: Icml '07: Proceedings of the 24th international conference on machine learning*. New York, NY, USA: ACM.

Wang, Muyuan, Li, Zhiwei, Lu, Lie, Ma, Wei-Ying, & Zhang, Naiyao. 2005. Web object indexing using domain knowledge. *Pages 294–303 of: Kdd '05: Proceedings of the eleventh acm sigkdd international conference on knowledge discovery in data mining*. New York, NY, USA: ACM.

Weigend, Andreas S., Wiener, Erik D., & Pedersen, Jan O. 1999. Exploiting hierarchy in text categorization. *Information retrieval*, **1**(3), 193–216.

Weiss, Sholom M., Apte, Chidanand, Damerau, Fred J., Johnson, David E., Oles, Frank J., Goetz, Thilo, & Hampp, Thomas. 1999. Maximizing text-mining performance. *Ieee intelligent systems*, **14**(4), 63–69.

Wermter, Stefan, & Hung, Chihli. 2002. Selforganizing classification on the reuters news corpus. *Pages 1–7 of: Proceedings of the 19th international conference on computational linguistics*. Morristown, NJ, USA: Association for Computational Linguistics.

Weston, J., & Watkins, C. 1998. *Multi-class support vector machines.* Tech. rept. CSD-TR-98-04. Department of Computer Science, Royal Holloway, University of London.

Wiener, Erik D., Pedersen, Jan O., & Weigend, Andreas S. 1995. A neural network approach to topic spotting. *Pages 317–332 of: Proceedings of SDAIR-95, 4th annual symposium on document analysis and information retrieval.*

Wu, Shih-Hung, Tsai, Tzong-Han, & Hsu, Wen-Lian. 2003. Text categorization using automatically acquired domain ontology. *Pages 138–145 of: Proceedings of the sixth international workshop on information retrieval with asian languages.* Morristown, NJ, USA: Association for Computational Linguistics.

Xu, Linli, & Schuurmans, Dale. 2005. Unsupervised and semi-supervised multi-class support vector machines. *Aaai.*

Yajima, Yasutoshi, & Kuo, Tien-Fang. 2006. Optimization approaches for semi-supervised multiclass classification. *Pages 863–867 of: Icdmw '06: Proceedings of the sixth ieee international conference on data mining - workshops.* Washington, DC, USA: IEEE Computer Society.

Yang, Hui, & Chua, Tat-Seng. 2004a. Effectiveness of web page classification on finding list answers. *Pages 522–523 of: Sigir '04: Proceedings of the 27th annual international acm sigir conference on research and development in information retrieval.* New York, NY, USA: ACM.

Yang, Hui, & Chua, Tat-Seng. 2004b. Web-based list question answering. *Page 1277 of: Coling '04: Proceedings of the 20th international conference on computational linguistics.* Morristown, NJ, USA: Association for Computational Linguistics.

Yang, Yiming. 1999. An evaluation of statistical approaches to text categorization. *Inf. retr.*, **1**(1-2), 69–90.

Yang, Yiming. 2001. A study of thresholding strategies for text categorization. *Pages 137–145 of: Sigir '01: Proceedings of the 24th annual international acm sigir conference on research and development in information retrieval.* New York, NY, USA: ACM.

Yang, Yiming, & Chute, Christopher G. 1994. An example-based mapping method for text categorization and retrieval. *Acm trans. inf. syst.*, **12**(3), 252–277.

Yang, Yiming, & Liu, Xin. 1999. A re-examination of text categorization methods. *Pages 42–49 of: Sigir '99: Proceedings of the 22nd annual inter-*

national acm sigir conference on research and development in information retrieval. New York, NY, USA: ACM.

Yang, Yiming, & Pedersen, Jan O. 1997. A comparative study on feature selection in text categorization. *Pages 412–420 of: Icml '97: Proceedings of the fourteenth international conference on machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Yang, Yiming, Slattery, Seán, & Ghani, Rayid. 2002. A study of approaches to hypertext categorization. *J. intell. inf. syst.*, **18**(2-3), 219–241.

Yarowsky, David. 1995. Unsupervised word sense disambiguation rivaling supervised methods. *Pages 189–196 of: Proceedings of the 33rd annual meeting on association for computational linguistics*. Morristown, NJ, USA: Association for Computational Linguistics.

Yu, Lei, & Liu, Huan. 2004. Efficient feature selection via analysis of relevance and redundancy. *J. mach. learn. res.*, **5**, 1205–1224.

Yuille, A. L., & Rangarajan, Anand. 2003. The Concave-Convex Procedure. *Neural computation*, **15**(4), 915–936.

Zhang, Dell, & Lee, Wee Sun. 2003. Question classification using support vector machines. *Pages 26–32 of: Sigir '03: Proceedings of the 26th annual international acm sigir conference on research and development in informaion retrieval*. New York, NY, USA: ACM.

Zhou, Dengyong, Schölkopf, Bernhard, & Hofmann, Thomas. 2004. Semi-supervised learning on directed graphs. *In: Advances in neural information processing systems*.

Zhou, Member-Zhi-Hua, & Li, Ming. 2005. Tri-training: Exploiting unlabeled data using three classifiers. *Ieee trans. on knowl. and data eng.*, **17**(11), 1529–1541.

Zhu, X., & Ghahramani, Z. 2002. *Towards semisupervised classification with markov random fields*.

Zhu, X., Ghahramani, Z., & Lafferty, J. 2003a. *Semi-supervised learning using gaussian fields and harmonic functions*.

Zhu, X., Lafferty, J., & Ghahramani, Z. 2003b. *Semisupervised learning: From Gaussian fields to Gaussian processes*.

Zhu, Xiaojin. 2005. *Semi-supervised learning with graphs*. Ph.D. thesis, Pittsburgh, PA, USA. Chair-John Lafferty and Chair-Ronald Rosenfeld.

Zien, Alexander, Brefeld, Ulf, & Scheffer, Tobias. 2007. Transductive support vector machines for structured variables. *Pages 1183–1190 of: Icml '07: Proceedings of the 24th international conference on machine learning.* New York, NY, USA: ACM.