Sistemas de Recuperación de Información

Aplicaciones del agrupamiento y de la clasificacin en la recuperación de información en la Web

---

Autores

Laura Victoria Riera Pérez

Marcos Manuel Tirador del Riego

Índice general

1. Agrupamiento

- Aprendizaje no supervisado
- Problema que resuelve

1.1. Algunas definiciones

- Flat clustering
- Hierarchical clustering
- Hard clustering
- Soft clustering
- Hipótesis de agrupamiento
- Cardinalidad

1.2. Flat clustering

- Medida de similitud:
- Medidas de evaluacion:
  - Criterio interno de calidad
  - Criterio interno de calidad
  - Pureza
  - Índice de frontera?
  - Medida F
- Algoritmos:
  - K-means
  - EM (generalización de K-means)

1.3. Hierarchical clustering

- Hierarchical agglomerative clustering
- Medidas de similitud:
  - Single link clustering
  - Complete link clustering
  - Centroid clustering
- Evaluación de calidad:
  - Group average link
  - Método de Ward
- Divisive clustering
- Cluster labeling
- Algoritmos:
  - Algoritmo HAC
  - Divisive Clustering

1.4. Aplicaciones a la RI

- Search result clustering
- Scatter-Gather
- Collection clustering
- Language modeling
- Cluster-based retrieval

1.5. Ventajas

1.6. Desventajas

2. Clasificación

El problema de clasificación en sentido general consiste en determinar dentro un conjunto de clases a cuá de ellas pertenece un objeto dado. En el marco de este documento estamos interesados en estudiar la clasificación de textos.

La forma más simple de clasificación de un conjunto de textos es la denominada clasificación en dos clases. Dichas clases están determinadas por un tópico específico y serían: *documentos sobre dicho tema* y *documentos no relacionados con el tema*. Este tipo de clasificación en ocasiones es llamada *filtrado*.

La forma más antigua de llevar a cabo la clasificación es manualmente. Por ejemplo, los bibliotecarios clasifican los libros de acuerdo a ciertos criterios, de modo que encontrar una información buscada no resulte una tarea de gigante dificultad. Sin embargo la clasificación manual tiene sus límites de escalabilidad.

Como alternativa podría pensarse el uso de *reglas* para determinar automáticamente si un texto pertenece o no a una colección de documentos relacionados con un tema. Por ejemplo las consultas permanentes son un ejemplo de regla aplicada automáticamente. Una consulta permanente es como una consulta normal pero que es ejecutada repetidamente sobre una colección de textos a la cual se van adicionando documentos nuevos constantemente. Su finalidad es determinar si los nuevos textos pertenecen o no a la clase en cuestión.

Una regla captura una cierta combinación de palabras claves que identifican una clase. Reglas codificadas a mano pueden llegar a ser altamente escalable, pero crearlas y mantenerlas requiere un elevado costo en recursos humanos.

Existe, sin embargo, un enfoque adicional a los dos anteriores mencionados. Nos referimos al uso de *Aprendizaje de Máquinas*. En este enfoque el conjunto de reglas de clasificación, o en general, el criterio usado para clasificar, es aprendido de forma automática a partir de los datos de entrenamiento.

Introduciremos a continuación la definición forma del problema de clasificación de textos, en el contexto del Aprendizaje de Máquinas.

Definition 1. *Sea* $\mathcal{X}$ *el espacio de documentos y* $\mathcal{C} := \{c_i \mid c_i \subset \mathcal{X}, i \in \{1, 2, \ldots, n\}\}$ *un conjunto fijo de clases (también llamadas categorías o etiquetas). Sea además* D *un conjunto entrenado de documentos clasificados* $(d, c) \in \mathcal{X} \times \mathcal{C}$*. El* problema de la clasificación de textos *consiste en encontrar, usando métodos o algoritmos de aprendizaje, una función* clasificadora $\gamma : \mathcal{X} \to \mathcal{C}$*, que mapee documentos a clases, que satisfaga que* $D \subset \gamma$*.*

El aprendizaje que toma parte en la búsqueda de γ es llamado *aprendizaje supervisado* debido a que se necesita la ayuda de uno o varios expertos que creen el conjunto de entrenamiento D. Estos expertos son también quienes determinan el conjunto de clases en que se clasificarán los textos. Denotaremos el método de aprendizaje supervisado descrito por Γ, el cual actúa como una función que mapea un conjunto de datos de entrenamiento en una función clasificadora, osea que $\Gamma(D) = \gamma$.

La definición dada en 1 implica que cada documento pertenece a una sola clase. Pero existe otro tipo de problemas que permiten que un documento pertenezca a más de una clase. Por ahora enfocaremos nuestra atención en el tipo de una clase.

2.1. Naive Bayes

Uno de los métodos más comunes de aprendizaje supervisado es el conocido como *Naive Bayes* (NB). Este es un método de aprendizaje probabilístico. La probabilidad de un documento d de pertenecer a una clase c se puede expresar como $P(c \mid d)$. La tarea del algoritmo es encontrar la mejor clase para cada documento d. Para ello NB establece que la clase más apropiada para un documento es la más probable, o sea

$$c_{map} = \arg\max_{c \in \mathcal{C}} P(c \mid d).$$

La clase escogida para d se denota por c_{map} debido a que este método de clasificación, de acuerdo a la clase más probable para un documento dado, es conocido como *maximum a posteriori* (MAP).

Sin embargo la probabilidad $P(c \mid d)$ es difícil de determinar. Haciendo uso del *Teorema de Bayes* podemos expresarla como

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}.$$

El factor de normalización $P(d)$ es usualmente ignorado ya que no aporta información a la hora de buscar la clase más apropiada para un documento d, ya que este tiene el mismo efecto en todos los candidatos. Este cálculo puede ser simplificado lo expresamos en términos de los términos en los documentos. Supongamos que $\{t_1, t_2, \ldots, t_n\}$ son los términos que aparecen en d. Entonces tenemos que

$$c_{map} = \arg\max_{c \in \mathcal{C}} P(c)P(d \mid c) = \arg\max_{c \in \mathcal{C}} P(c) \prod_{1 \leq k \leq n} P(t_k \mid c),$$

donde $P(t_k \mid c)$ es la probabilidad de que el término t_k aparezca en un documento de la clase c. Podemos considerar $p(t_k \mid c)$ como una medida de qué tanto demuestra el término t_k que c es la clase correcta. El término $P(c)$ es conocido como probabilidad previa (*prior probability*) y en caso de que la información

aportada por los términos no sea determinante en la selección podemos siempre escoger la clase con mayor valor de $P(c)$.

Para simplificar aún mas el cómputo podemos sustituir los valores anteriores por sus logaritmos. Esto reducir'a el costo de hacer los cálculos y además los errores aritméticos dado que la multiplicación se transforma en suma. La clase seleccionada sería entonces

$$c_{map} = \arg\max_{c \in \mathcal{C}} \left(\log(P(c)) + \sum_{1 \leq k \leq n} \log(P(t_k \mid c)) \right).$$

Solo nos queda ver como estimamos los parámetros $P(c)$ y $P(t_k \mid c)$ dado que los valores reales no son posibles de calcular. Para la probabilidad previa podemos contar la frecuencia relativa de cada clase en D:

$$P(c) = \frac{N_c}{N},$$

donde N_c es el número de documentos en la clase c y N es el numero total de documentos. Procedemos de manera similar para la probabilidad específica de una palabra en una clase

$$P(t_k \mid c) = \frac{T_{c,t_k}}{T_c},$$

donde T_{c,t_k} indica la cantidad de veces que ocurre la palabra t_k en todos los documentos de la clase c y T_c es la cantidad total de palabras contando repeticiones) en toda la clase c. Si embargo, aún tenemos un problema con estas fórmulas y es que estamos asignando probabilidad cero a todos las clases que no contengan a todas las palabras del documento a clasificar. Para evitar esto adicionamos por defecto una unidad a cada contador lo cual es conocido como *Laplace smoothing*

$$P(t \mid c) = \frac{T_{c,t} + 1}{T_c + |V|},$$

donde $|V|$ es el número total de téminos en el vocabulario.

Es importante destacar que en este método estamos obviando la posición de las palabras. Presentamos aquí los algoritmos para entrenar y clasificar usando BN que fueron textualmente copiados de 2009 Manning C. D., Introduction to Information Retrieval, página 260. [No recuerdo como citar esto correctamente. Also recordar que agregue unos paquetes arriba que no se si se agragan aqui o en otro de los .tex].

Podemos deducir de los algoritmos que la complejidad de ambos es linear en el tiempo que toma escanear la información. Dado que esto hay que hacerlo al menos una vez, se puede decir que este método tiene complejidad temporal óptima. Dicha eficiencia hace que NB sea un método de clasificación tan usado.

- Aprendizaje supervisado
- Problema que resuelve

- Rule-based classification

Algorithm 1 TrainMultinomial

Require: Set of classes $\mathcal{C}$ and training set D.

$V \leftarrow ExtractVocabulary(D)$
$N \leftarrow CountDocs(D)$
for $c \in \mathcal{C}$ **do**
    $N_c \leftarrow CountDocsInClass(D, c)$
    $prior[c] \leftarrow N_c/N$
    $text_c \leftarrow ConcatenateTextOfAllDocsInClass(D, c)$
    for $t \in V$ **do**
        $T_{ct} \leftarrow CountTokensOfTerm(text_c, t)$
    for $t \in V$ **do**
        $condprob[t][c] \leftarrow \frac{T_{c,t}+1}{\sum_{t'}(T_{c,t'}+1)}$
return $V, prior, condprob$

Algorithm 2 ApplyMultinomialNB

Require: $\mathcal{C}, V$, $prior$, $condprob$, d

$W \leftarrow ExtractTokensFromDoc(V, d)$
for $c \in \mathcal{C}$ **do**
    $score[c] \leftarrow \log prior[c]$
    for $t \in W$ **do**
        $score[c] += \log condprob[t][c]$
return $\arg\max_{c \in \mathcal{C}}(score[c])$

- Statistical classification
- Feature selection
- Medidas de evaluación:
  - Fitting
  - Precisión
  - Recobrado
  - Medida F (balanceada)
  - Classification accuracy
- Algoritmos:
  - Naive Bayes
  - K-Nearest Neighbours

2.2. Aplicaciones a la RI

- Standing queries
- Spam filtering

2.3. Ventajas

2.4. Desventajas

3. Agrupamiento vs. Clasificación

4. Ejemplos de aplicacin