Clasificación de pacientes infecciosos de tuberculosis mediante métodos núcleo basados en la divergencia de Kullback-Leibler

J. Montoya Martínez[1], A. Artés Rodríguez[1]

[1] Departamento de Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid, Leganés, España, {jmontoya,antonio}@tsc.uc3m.es

Resumen

En este artículo se considera el problema de clasificación de pacientes infecciosos de tuberculosis a partir del análisis y procesado de las salidas blandas de clasificadores locales que determinan la presencia o ausencia de bacilo en parches de imágenes del esputo del paciente. Para atacar este problema se propone una nueva versión del núcleo de Kullback-Leibler, el cual puede ser aplicado a un conjunto de muestras de funciones de densidad de probabilidad (fdp) sin necesidad de modelar explícitamente la distribución de los datos. Nuestra aproximación se basa en trabajar directamente sobre las muestras usando un estimador de la divergencia de Kullback-Leibler (KL) para calcular el valor del núcleo. Finalmente se muestra las prestaciones de un clasificador SVM con este núcleo en una base de datos de imágenes de esputo tinturado con auramina.

1. Introducción

Una técnica tradicional para el diagnóstico de la tuberculosis es la baciloscopia de esputo, la cual consiste en la visión directa mediante microscopio del bacilo de Koch en esputo, empleando una tinción fluorescente de auramina. Los métodos usados para automatizar este proceso se concentran en la búsqueda de bacilos en la imagen microscópica del esputo. En [1], se utiliza un método basado en la fusión de las salidas blandas de clasificadores locales que determinan la presencia o ausencia de bacilos en parches de las imágenes de esputo del paciente. Esta fusión se hace a través del modelado estadístico de dichas salidas y su respectivo uso en un test secuencial que determina si un paciente es infeccioso o no. La propuesta que se presenta en este artículo ataca el mismo problema, pero sin la necesidad de crear explícitamente un modelo probabilístico de los datos. Nuestra propuesta es un método máquina que trabaja directamente sobre las salidas blandas, procesandolas con una nueva versión del núcleo de Kullback-Leibler, de tal manera que luego son clasificadas en un espacio de características de dimensión infinita usando máquinas de vectores soporte (SVM).

Los componentes fundamentales de la aproximación que proponemos para atacar este problema de clasificación son las máquinas de vectores soporte y los núcleos probabilísticos. Las máquinas de vectores soporte son excelentes representantes de los clasificadores basados en modelos discriminantes, han sido usadas en aplicaciones de diversos campos (reconocimiento de texto, imágenes, video y voz, entre otros), logrando excelentes resultados [2]. Sin embargo, es bien sabido que no podemos introducir conocimiento a priori en ellas (en forma de modelos probabilísticos de los datos), y además, se les dificulta trabajar con secuencias de datos de diferente longitud. ¿Cómo podemos fortalecer tales debilidades de las SVM? Esto es precisamente lo que se busca con los núcleos probabilísticos. Para lograrlo, se hace que el núcleo sea función de las fdp de los datos.

El campo de investigación de los núcleos probabilísticos comenzó con la aparición del núcleo de Fisher [3]. Este núcleo permite medir distancias entre datos calculando las características dado un modelo probabilístico $p(x|\theta)$. Primero se realiza una estimación del parámetro θ a partir de los datos de entrenamiento, dando como resultado $\hat{\theta}$. Luego, el vector tangente de la log-verosimilitud marginal $\log(P(x|\hat{\theta}))$ es usado como un vector de características. El núcleo de Fisher se refiere al producto interno en este espacio de características. Este núcleo ha sido combinado con las SVM, logrando obtener excelentes resultados de clasificación en varios campos, como por ejemplo en el análisis de ADN y proteínas [3].

Otro representante importante de los núcleos probabilísticos es el núcleo TOP [4]. Mientras que el núcleo de Fisher se calcula a partir de la log-verosimilitud marginal, el núcleo TOP se deriva a partir de los vectores tangentes de la log-probabilidad a posteriori

$$\upsilon(x,\theta)=\log(y_{c_1}|x,\theta)-\log(y_{c_{-1}}|x,\theta)$$

y se define como:

$$k(x,z)=f_\theta(x)^T f_\theta(z)$$

siendo $f_\theta(x)$

$$f_\theta(x)=[\upsilon(x,\theta),\partial_{\theta_1}\upsilon(x,\theta),\ldots,\partial_{\theta_N}\upsilon(x,\theta)]^T$$

Como hemos visto, tanto en el núcleo de Fisher como en el núcleo TOP se trabaja con modelos probabilísticos que representan a todos los datos. Un núcleo probabilístico que se diferencia de los dos anteriores, en cuanto a que cada dato se modela estadísticamente de manera indivual, es el núcleo de Kullback-Leibler [5]. En este artículo se propone una versión de este núcleo que permite trabajar directamente con un conjunto de estimas muestrales de fdp continuas sin necesidad de construir explícitamente un modelo de las distribuciones de los datos. Para esto, hacemos uso de una estimación de la divergencia KL que se basa en el algoritmo de *k*-vecinos próximos. Al final,

veremos que con esta aproximación sólo necesitamos la distancia de cada muestra a su vecino más próximo.

El resto del artículo está organizado de la siguiente manera: En la sección 2 se muestra la definición del núcleo de Kullback-Leibler dada en [5]. En la sección 3 se describe el modelo propuesto, el cual se basa en la utilización de una aproximación de vecinos próximos para estimar la divergencia KL. En la sección 4 se analiza las prestaciones del núcleo propuesto usándolo en una aplicación de imágenes médicas relacionada con el diagnóstico de tuberculosis. Finalmente en la sección 5 se presentan las conclusiones de este artículo.

2. Núcleo de Kullback-Leibler

Es bien conocido que el componente principal de las máquinas de vectores soporte es el núcleo [2], cuya tarea principal es la de capturar la similitud entre datos. Dado que los datos son naturalmente descritos por sus densidades de probabilidad, una idea que ha recibido mucha atención es la de reemplazar estos por sus respectivas fdp [3,4]. Por lo tanto, es razonable desear una métrica que compare directamente fdp. Entre las muchas posibilidades que se conocen de los campos de la estadística y la teoría de la información, en este artículo expondremos resultados alcanzados usando la divergencia simétrica de Kullback-Leibler, la cual se define como [6]:

$$J(f_i, f_j) = KL(f_i(x), f_j(x)) + KL(f_j(x), f_i(x)) \quad (1)$$

donde $f_i(x)$ y $f_j(x)$ son fdp que modelan a un dato *i* y a un dato *j* respectivamente, y $KL(f_i(x), f_j(x))$ es la divergencia de Kullback-Leibler entre las densidades $f_i(x)$ y $f_j(x)$ [6]:

$$KL(f_i(x), f_j(j)) = \int_{\mathbb{R}^D} f_i(x) \log\left(\frac{f_i(x)}{f_j(x)}\right) dx \geq 0 \quad (2)$$

Debido a que una matriz de núcleo basada directamente en (1) no satisface las condiciones de Mercer [2] (no es definida positiva), se necesita realizar un paso adicional para definir un núcleo válido. Entre las muchas alternativas posibles, en [5] propusieron la siguiente versión del núcleo de Kullback-Leibler (KLK):

$$KLK(f_i(x), f_j(x)) = e^{-aJ(f_i(x), f_j(x))+b} \quad (3)$$

Como se puede observar, el núcleo KL es muy flexible y puede ser adaptado al problema de clasificación de varias maneras, ya sea a través del modelado estadístico de la base de datos bajo consideración o usando diversas aproximaciones de (2).

3. Núcleo de Kullback-Leibler usando un estimador K-NN

En este artículo se propone una versión del núcleo KL que puede ser aplicado a un conjunto de estimas muestrales de fdp continuas sin la necesidad de construir explícitamente un modelo probabilístico de las distribuciones de los datos. Se basa en un estimador de vecinos próximos para estimar la divergencia KL, cuyas propiedades de convergencia han sido demostradas en [7,8].

Se tienen dos conjuntos *P y Q con n y m* muestras i.i.d de los datos *i y j respectivamente*

$$P = \left\{x_p^{(i)}\right\}_{p=1}^{n}, Q = \left\{x_q^{(j)}\right\}_{q=1}^{m}$$

siendo $x_p^{(i)}$ la muestra *p*-ésima del dato *i* y $x_q^{(j)}$ la muestra *q*-ésima del dato *j*. Podemos usar integración Monte-Carlo y estimación mediante *k*-vecinos próximos para aproximar (2) de la siguiente manera:

$$\widehat{KL}(f_i(x), f_j(x)) = \frac{1}{n}\sum_{p=1}^{n} \log \frac{\hat{f}_i(x_p^{(i)})}{\hat{f}_j(x_q^{(j)})} \quad (4)$$

siendo $\hat{f}_i(x_p^{(i)})$ y $\hat{f}_j(x_q^{(j)})$ la estimación por *k*-vecinos próximos de $f_i(x)$ y $f_j(x)$ respectivamente.

Según experimentos realizados en [7], la estimación de (4) para $k = 1$ es la que converge más rápidamente al valor verdadero. Por lo tanto, $\hat{f}_i(x_p^{(i)})$ y $\hat{f}_j(x_q^{(j)})$ son las estimaciones de f_i y f_j respectivamente, usando el vecino más próximo. Lo anterior implica que (4) pueden ser encontrada de la siguiente manera:

$$\widehat{KL}(f_i, f_j) = \frac{D}{n}\sum_{p=1}^{n} \log \frac{d_Q(x_p^{(i)})}{d_P(x_p^{(i)})} + \log \frac{m}{n-1} \quad (5)$$

donde $d_P(x_p^{(i)})$ es la distancia euclídea de la muestra *p*-ésima del dato *i* a su vecino más próximo en el conjunto *P*, $d_Q(x_p^{(i)})$ es la distancia euclídea de la muestra *p*-ésima del dato *i* a su vecino más próximo en el conjunto *Q*.

De esta manera, el núcleo de Kullback-Leibler usando la aproximación de vecinos próximos queda expresado como:

$$KLK(f_i, f_j) = e^{a(\widehat{KL}(f_i, f_j) + \widehat{KL}(f_j, f_i)) + b} \quad (6)$$

Analicemos el problema dual de optimización de una SVM teniendo en cuenta nuestro núcleo:

$$\begin{aligned} &\hat{\alpha} = argmin \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j KLK(f_i, f_j) - \sum_{i=1}^{l} \alpha_i \\ &\textit{sujeto a} \\ &0 \leq \alpha_i \leq C, \forall\, i = 1, \ldots, l \\ &\sum_{j=1}^{l} \alpha_j y_j = 0 \end{aligned} \quad (7)$$

Para resolver eficientemente este problema evaluamos de manera offline $KLK(f_i(x), f_j(x))$. Para esto, debemos calcular dos vectores para cada dato *i* del conjunto de entrenamiento: Un vector que contiene las distancias a los vecinos próximos de cada muestra *p* del dato *i* respecto al conjunto de muestras i.i.d del dato *i y u*n vector que contiene las distancias a los vecinos próximos de cada muestra *p* del dato *i* respecto al conjunto de muestras i.i.d del dato *j*, $\forall\, i, j = \backslash, \ldots, l$, siendo *l* el número de datos de entrenamiento. Existen técnicas algorítmicas para calcular estos vectores de manera eficiente [9].

Para evaluar la función de decisión de la SVM ante una entrada nueva

$$D(x) = signo\left(\sum_{i \in SVs} \alpha_i y_i KLK(f_i, f_j) + b\right)$$

calculamos lo siguiente: Un vector que contiene las distancias a los vecinos próximos de cada muestra *p* del dato *i* respecto al conjunto de muestras i.i.d de la nueva

entrada , un vector que contiene las distancias a los vecinos próximos de cada muestra i.i.d de la nueva entrada respecto al conjunto de muestras i.i.d del dato *i y u*n vector que contiene las distancias a los vecinos próximos de cada muestra i.i.d de la nueva entrada respecto a dicho conjunto, $\forall i \in SVs$, siendo SVs el conjunto de vectores soporte.

4. Clasificación de pacientes

Este núcleo lo vamos a aplicar al problema descrito anteriormente. La base de datos que se usará en la pruebas es la misma que se usó en [1] , la cual fue generada de la siguiente manera: Cada una de las imágenes es dividida en pequeñas regiones que se solapan, y cada una de estas regiones es presentada a un clasificador local, el cual se encarga de decidir la presencia o ausencia de bacilo. La base de datos consiste en las salidas blandas de este clasificador. En [1], dichas salidas blandas son enviadas a un centro de fusión, en el cual se modelan estadísticamente. Este modelo es luego usado en un test secuencial, el cual determina si un paciente es infeccioso o no. Nosotros procesamos nuevamente esta base de datos de tal manera que para cada uno de los pacientes se formó un vector de muestras con todas las salidas blandas de sus respectivas imágenes. En nuestra aproximación partimos directamente del conjunto de las estimas muestrales (las salidas blandas) sin necesidad de construir explícitamente un modelo de las distribuciones de los datos, y les aplicamos nuestra versión propuesta del núcleo KL. Todas las imágenes corresponden a un total de 46 pacientes, 35 no infecciosos y 11 infecciosos, y por tanto tenian bacilos en el esputo.

Se utilizó la misma división del conjunto de datos para entrenamiento y para test que la usada en [1], 20 pacientes sanos y 9 pacientes enfermos para entrenamiento, y 15 pacientes sanos y 2 pacientes enfermos para test.

Debido a que la base de datos a procesar era desbalanceada se usaron diferentes clases de máquinas de vectores soporte para buscar dar solución a las malas prestaciones que las SVM tienen con este tipo de bases de datos. Los resultados obtenidos en el conjunto de test se muestran a continuación y se comparan con los obtenidos en [1]:

Clasificador	Sens.	Espec.
Weighted SVM (KL-KNN)	0.5	0.73
SVM-WHM (KL-KNN)	0.5	1
SVM *CV-F* (KL-Histograma)	1	0.6
SVM *CV-F* (KL-Monte-Carlo)	1	0.8
SVM *CV-F* (KL-KNN)	1	1
Test Secuencial	1	1

***Tabla 1**. Sensibilidad y Especificidad de los clasificadores implementados.*

En primer lugar, teniendo en cuenta el conjunto de test usado (15 pacientes sanos y 2 pacientes enfermos), si clasificamos mal a un paciente enfermo nuestra sensibilidad baja a 0.5. Si clasificamos mal a 3, 4 y 6 pacientes sanos nuestra especificidad baja a 0.8, 0.73 y 0.6 respectivamente. El primer modelo que se probó fue el Weighted SVM [10]. Este consiste en asignarle un peso diferente a cada una de las clases a clasificar de acuerdo a la siguiente proporción:

$$\frac{w_p}{w_n} = \frac{l_n}{l_p}, w_p + w_n = 1$$

siendo *lp* y *ln* el número de ejemplos de la clase +1 y -1 respectivamente. Una vez fijados los pesos de acuerdo al conjunto de entrenamiento, los parámetros *C* en (7), correspondientes a cada una de las clases son: $C_{+1} = C \cdot w_p$, $C_{-1} = C \cdot w_n$.

Los resultados obtenidos con este modelo se muestran en la tabla (1) (se utilizó validación cruzada para obtener el parámetro *C*). Como se puede observar, es el que tuvo peores prestaciones de los modelos SVM implementados. Esto se debe a que dicho modelo asume que los datos están distribuidos de manera uniforme por todo el espacio, y cuando no es así, la proporción de las clases, y por tanto la de los pesos, no puede describir el desbalance de los vectores soporte de manera correcta.

El segundo modelo que se probó fue el SVM-WHM (Support Vector Classifier with Weighted Harmonic Mean Offset) [11]. En este modelo primero se entrena al clasificador SVM de manera estándar para conseguir el conjunto de vectores soporte y el parámetro *C* (se usa validación cruzada para encontrar el valor de este último). Luego se modifica la función de decisión de la siguiente manera:

$$D(x) = signo\left(\sum_{i \in SVs} \alpha_i y_i K(x_i, x) + b - \delta \right)$$

donde δ es la media armónica ponderada de las salidas blandas de la SVM cuando las entradas son sus respectivos vectores soporte. Usando este modelo conseguimos mejorar los resultados del modelo Weighted SVM, mejoramos la especificidad al lograr clasificar correctamente a todos los pacientes sanos, pero no logramos mejorar su sensibilidad ya que se sigue clasificando mal a uno de los pacientes enfermos.

En los anteriores dos modelos los parámetros de la SVM se calcularon usando validación cruzada estándar, en la cual la función que mide las prestaciones de la máquina es el error de clasificación. Esta medida no tiene en cuenta el desbalance que puedan tener los datos de entrenamiento. Para tratar de mejorar esto, se utiliza en la validación cruzada una función diferente al error de clasificación, en nuestro caso se utilizó una función conocida con el nombre de medida *F* [12], en la cual se define una ponderación entre la sensibilidad y la precisión:

$$F = \frac{2 \cdot pr \cdot se}{pr + se}$$

siendo *pr* y *se* la precisión y la sensibilidad del clasificador respectivamente. El modelo SVM con CV-F (validación cruzada usando la medida *F*) logra clasificar perfectamente a todos los pacientes, y por tanto, de los tres clasificadores SVM implementados usando el núcleo

propuesto es el que mejores prestaciones tiene. A fin de comparar el núcleo propuesto con otras versiones del núcleo de Kullback-Leibler se implementaron dos clasificadores SVM, uno usando la versión del núcleo KL basado en histogramas (KL-Histograma) y el otro usando la versión del núcleo KL basado en una aproximación Monte-Carlo (KL-Monte-Carlo) (en cada uno de ellos se utilizó la validación cruzada usando la medida F).

Para la implementación del clasificador SVM basado en KL-Histograma primero se modelo la distribución de cada uno de los datos de manera no paramétrica a través de histogramas, luego se calculó la divergencia KL entre histogramas [5] y finalmente se evaluo el núcleo usando (6). Este clasificador obtuvo unas prestaciones inferiores respecto al mejor clasificador obtenido basado en el núcleo propuesto ya que obtuvo una especificidad mucho menor (0.6). Además este núcleo debe evaluar de manera explícita la distribución de cada uno de los datos, paso no necesario en el núcleo propuesto.

Para la implementación del clasificador SVM basado en KL-Monte-Carlo primero se modelo la distribución de cada uno de los datos de manera paramétrica a través del uso de mezcla de gaussianas. Para escoger el número óptimo de componentes en la mezcla se utilizó una selección de modelo bayesiana. Luego, se calculó la divergencia KL siguiendo su respectiva expresión cuando las fdp son miembros de una familia exponencial [5] y finalmente se evaluo el núcleo usando (6). Este clasificador obtuvo unas mejores prestaciones respecto al clasificador basado en KL-Histograma, pero aún así fue inferior respecto al mejor clasificador obtenido basado en el núcleo propuesto ya que obtuvo una especificidad menor (0.8). Además este núcleo es más costoso computacionalmente ya que debe calcular de manera explícita la distribución paramétrica de cada uno de los datos, paso no necesario en el núcleo propuesto.

El test secuencial implementado en [1] logra clasificar correctamente a todos los pacientes, pero esta aproximación es compleja y costosa computacionalmente debido a que también hace uso de diversas técnicas como ventanas de Parzen y bootstrap. Además, es poco robusto, ya que una variación pequeña en los parámetros del test puede conducir a un incremento notorio en el número de muestras necesarias para que el test converja. Nuestra mejor aproximación (usando SVM con CV-F) logra unas prestaciones comparables al anterior y es mucho más sencilla y eficiente, ya que el núcleo propuesto tiene implícitamente en cuenta las distribuciones de los datos y sólo necesita las distancias a los vecinos más próximos de cada muestra, esto se puede calcular de manera rápida y eficiente, lo cual se traduce en un núcleo computacionalmente eficiente.

5. Conclusiones

En este artículo se ha analizado una nueva herramienta para atacar el problema de clasificación de pacientes bacilíferos infecciosos basada en métodos núcleo y teoría de la información. Esta herramienta es una nueva versión del núcleo de Kullback-Leibler. Puede ser aplicada a un conjunto de estimas muestrales de fdp continuas sin la necesidad de construir explícitamente un modelo probabilístico de los datos, puede usarse en datos de diferente longitud (diferente número de muestras) y es una herramienta computacionalmente eficiente ya que sólo requiere las distancias a los vecinos más próximos de cada muestra. Finalmente se mostró su uso en una base de datos de imágenes de esputo tinturado con auramina, logrando muy buenas prestaciones.

Agradecimientos

Queremos expresar nuestros agradecimientos al Dr. Ricardo Santiago Mozos por haber facilitado la base de datos utilizada en las pruebas. Este trabajo ha sido parcialmente financiado por el Ministerio de Educación (proyectos 'DEIPRO', id. TEC2009-14504-C02-01, y 'COMONSENS', id. CSD2008-00010).

Referencias

[1] Santiago-Mozos R, Artés-Rodríguez A. Uncertainty-based Censoring Scheme in Distributed Detection Using Learning Techniques. *Proceedings of the Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, 2006, pp. 2027–2034.

[2] Schölkopf B, Smola A. Learning with kernels. MIT Press, Cambridge,MA, USA, 2001, (ISBN 0-262-19475-9).

[3] Jaakkola T, Diekhans M, Haussler D. Using the Fischer Kernel Method to Detect Remote Protein Homologies. *Proc. of the International Conference on Intelligence Systems for Molecular Biology*, 1999.

[4] Tsuda K, Kawanabe M, Ratsch G, Sonnenburg S, K. Muller. A New Discriminative Kernel from Probabilistic Models. *Neural Computation*, vol. 14, no. 10, 2002.

[5] Moreno P.J, Ho P.P, Vasconcelos N. A Kullback-Leibler Divergence Based Kernel for SVM Classification in Multimedia Applications. *Proc. of NIPS*, 2003.

[6] Cover T. M, Thomas J. A. Elements of Information Theory. John Wiley & Sons, 2a edición, 2006 (ISBN 0-471-24195-4).

[7] Pérez-Cruz F. Estimation of Information Theoretic Measures for Continuous Random Variables. *Proc. of NIPS*, 2008.

[8] Wang Q, Kulkarni S, Verdú S. A Nearest-Neighbor Approach to Estimating Divergence Between Continuous Random Vectors. *IEEE Int. Symp. Information Theory*, Seattle, USA, 2006.

[9] Kim B.S, Park S.B. A Fast K Nearest Neighbor Finding Algorithm Based on the Ordered Partition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, 1986.

[10] Du S.-X, Chen S.-T. Weighted Support Vector Machine for Classification. *IEEE Int. Conf. on Systems, Man and Cybernetics*, 2005.

[11] Li B, Hu J, Hirasawa K. Support Vector Machine Classifier with WHM Offset for Unbalanced Data. *Journal of Advanced Computational Intelligence and Intelligence Informatics*, vol. 12, no. 1, 2008.

[12] Eitrich T, Lang B. Efficient Optimization of Support Vector Machine Learning Parameters for Unbalanced Datasets. *Journal of Computational and Applied Mathematics*, vol. 196, issue 2, pp.425-436, 2006.