Análisis de estrategias de clasificación multiclase en microarrays: relación con medidas de complejidad

L. Morán Fernández, V. Bolón-Canedo y A. Alonso-Betanzos

Laboratorio de I+D en Inteligencia Artificial (LIDIA), Departamento de Computación, Universidade da Coruña, Campus de Elviña s/n, A Coruña, 15071, España

Resumen Existen varias aproximaciones para enfrentarse a problemas de aprendizaje que involucran a más de dos clases, entre ellos la aplicación directa de algoritmos 'multiclase' o la división del problema original en varios subproblemas de clasificación binaria. Dentro de los esquemas que pueden ser empleados para generar problemas binarios a partir de un conjunto de datos con múltiples clases, en este trabajo utilizaremos la técnica *one-versus-one*, donde se emplearán cuatro métodos diferentes para combinar los resultados de los clasificadores binarios. Se compararán los resultados de clasificación obtenidos por ambas aproximaciones sobre diez conjuntos de datos microarray con un número de clases entre tres y once, antes y después de aplicar varios métodos de selección de características. Además, se analizará en profundidad la complejidad teórica de estos conjuntos haciendo uso de medidas de complejidad, para luego conectarlos con los resultados empíricos obtenidos por las diferentes estrategias de clasificación.

1. Introducción

Tradicionalmente, se han aplicado diversas estrategias de descomposición para lidiar con problemas de clasificación multiclase. La forma más simple de clasificar problemas que involucran más de dos clases consiste en utilizar un clasificador multiclase. Sin embargo, no es una elección que pueda tomarse en todos los casos, ya que no todos los algoritmos de aprendizaje tienen esa capacidad. Además, se ha observado que la aplicación directa de una estrategia de aprendizaje para un problema de múltiples clases puede dar como resultado un *sobreajuste* en aquellas clases que aparecen en mayor proporción en el conjunto de datos o bien que son fácilmente separables [6]. Por otra parte, Guyon y Elisseeff [8] defienden que cuanto mayor es el número de clases menos probable resulta que un conjunto único de características proporcione una buena separación. Otra aproximación consiste en dividir el problema original en múltiples problemas binarios y realizar la clasificación mediante el entrenamiento y la combinación de varios clasificadores binarios. La mayoría de estas estrategias están incluidas en el marco *Error-Correcting Output Codes* (ECOC). Entre ellos, el esquema *One-vs-One* es una de las técnicas más utilizadas. Su uso es frecuente en aplicaciones del mundo real, siendo una forma simple aunque efectiva de manejar este tipo de problemas con múltiples clases. Estos problemas multiclase son divididos en subproblemas binarios, los cuales aprenden por diferentes clasificadores cuyas salidas se combinan para clasificar una muestra. No obstante, esta aproximación también tiene sus inconvenientes,

como puede ser integrar la información que proviene de los múltiples clasificadores binarios o el hecho de que exista una representación suficiente de cada una de las clases a tratar dentro del conjunto de entrenamiento.

Para realizar las diferentes pruebas de experimentación hemos utilizado conjuntos microarray. Este tipo de datos es utilizado para recopilar información de tejidos y células observando diferencias en la expresión genética con el fin de facilitar el diagnóstico de enfermedades o distinguir entre tipos específicos de tumores. La clasificación de estos conjuntos de datos supone un gran desafío para las técnicas computacionales debido al elevado número de características y el pequeño tamaño muestral. Además, un problema común en microarrays es el denominado *desbalanceo de clases*. Esto ocurre cuando un conjunto de datos está dominado por una o varias clases mayoritarias que tienen un número significativamente mayor de muestras que las clases menos frecuentes/minoritarias en el conjunto. En este caso, los algoritmos de clasificación tienen un sesgo hacia las clases con un mayor número de muestras, ya que las reglas que predicen correctamente los casos ponderan positivamente a favor de la precisión métrica, mientras las reglas que predicen ejemplos de la clase minoritaria son generalmente ignorados (tratados como ruido). Todas estas características convierten a los microarray en unos conjuntos de datos idóneos para realizar este tipo de experimentos.

Tratando de analizar en profundidad el rendimiento de las diferentes estrategias de clasificación, haremos uso de las medidas de complejidad propuestas por Ho y Basu [12] para conectar la complejidad teórica de cada uno de los conjuntos con los resultados de clasificación. Por otra parte, y debido al elevado número de características que se manejan, resulta interesante la aplicación de diferentes métodos de selección de características, ya que varios estudios muestran que la mayoría de genes tomados en experimentos con microarrays no son relevantes para una distinción precisa entre las diferentes clases en las que se divide el problema [7].

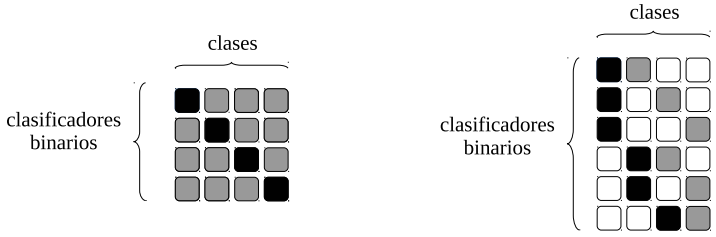
El presente artículo está organizado de la siguiente forma: la Sección 2 explica las diferentes estrategias de clasificación para problemas con múltiples clases, así como los cuatro métodos de decodificación utilizados. En la Sección 3 se especifican los conjuntos de datos empleados, las medidas de complejidad, los algoritmos de clasificación y los filtros para reducir la alta dimensionalidad. Los resultados experimentales se presentan en la Sección 4. Por último, en la Sección 5 se analizan las conclusiones y además, se indican algunas líneas de trabajo futuro.

2. Aproximación mediante múltiples clasificadores binarios

Dado un problema de clasificación con c clases, siendo $c > 2$, un método estándar para realizar la conversión a múltiples problemas de clasificación binarios es tratar un subconjunto de clases como ejemplos positivos y el conjunto de clases restantes como ejemplos negativos. Esto se repite para distintos subconjuntos de clases y se construye un clasificador/modelo para cada uno. De esta forma, se obtienen l subconjuntos de entrenamiento binarios. Una vez entrenados los modelos, es necesario un mecanismo para integrar los resultados de los l clasificadores binarios y obtener un resultado final para el problema original multiclase [20]. Los distintos procesos de integración fueron generalizados en el método ECOC [5].

En este método, en lugar de proporcionarle a cada algoritmo $s = 1, ..., l$ los datos etiquetados (x_i, y_i), la salida deseada y_i se transforma de acuerdo a una matriz codificada $M_{l \times c}$, de manera que el algoritmo de aprendizaje recibe como conjunto de entrenamiento los pares $(x_i, M(s, y_i))$. La matriz M depende del esquema escogido para generar los subconjuntos binarios, siendo los más habituales los siguientes [13,14]:

- *One-versus-rest* (OVR): es la técnica más popular, y consiste en tomar una clase y aprender a discriminar esa clase del resto. Transforma un problema de c clases en c problemas binarios, de forma que $l = c$. Estos problemas de dos clases se construyen usando los ejemplos de la clase i como ejemplos positivos y los ejemplos del resto de clases como los ejemplos negativos. La matriz codificada M para esta técnica se puede ver en la Figura 1(a), y consta de tantas filas como etiquetas de clase tuviera el problema. Los cuadrados negros representarían la clase "one" y los cuadrados grises "rest".
- *One-versus-one* (OVO): la idea de esta técnica es muy simple, consiste en entrenar un clasificador para cada par de clases. Transforma un problema de c clases en $c(c-1)/2$ problemas binarios $< i, j >$, uno por cada conjunto de clases i, j, donde $i, j = 1, ..., c$ e $i < j$. El clasificador binario para un caso $< i, j >$ es entrenado con clases i y j, mientras que las muestras de las clases $k \neq i, j$ son ignoradas. La matriz codificada M para esta técnica sería la que se puede observar en la Figura 1(b). Las etiquetas positivas están representadas como cuadrados negros, las negativas como cuadrados grises y los cuadrados blancos indican las etiquetas de clases que no se emplean.



(a) One-versus-rest (b) One-versus-one

Figura 1. Matriz codificada para un problema con cuatro clases

Una vez establecidas la matriz de codificación M y los distintos subconjuntos, el algoritmo de aprendizaje binario se usa para construir clasificadores, uno para cada columna de M. Este conjunto es considerado el conjunto de entrenamiento por el algoritmo de aprendizaje, que busca una hipótesis h_s. Esta reducción produce l clasificadores binarios $h_1, ..., h_l$. El vector de predicciones de estos clasificadores sobre una muestra x se denota como $h(x) = (h(x), ..., h_l(x))$ y la *r-ésima* fila de M como M_r. Dada una nueva muestra x, se predice la etiqueta y que minimiza $d(M_y, h(x))$ para cualquier distancia d. Para obtener la salida apropiada existen diferentes técnicas:

- *Hamming:* esta técnica cuenta el número de posiciones s en las cuales el signo de la predicción $h_s(x)$ difiere de la entrada de la matriz $M(r,s)$. La medida de la distancia es:

$$d_H(M_r, h(x)) = \sum_{s=1}^{l} \left(\frac{1 - sign(M(r,s)h_s(x))}{2} \right)$$

  donde $sign(z)$ es $+1$ si $z > 0$, -1 si $z < 0$ y 0 si $z = 0$. Para una muestra x y una matriz M, la etiqueta de clase predicha $y \in 1, ..., c$ es:

$$\hat{y} = arg \min_r d_H(M_r, h(x))$$

- *Loss-based:* este método considera la magnitud de las predicciones, la cual puede ser con frecuencia una indicación del nivel de confianza, así como también de la relevancia de la *función de pérdida L*. La idea es escoger la etiqueta r que sea más consistente con las predicciones $h_s(x)$, en el sentido de que la muestra x está etiquetada como de la clase r y la *pérdida* total sobre la muestra (x, r) es reducida al mínimo sobre las opciones de $r \in 1, ..., c$. Esto significa que esta medida de la distancia es la pérdida total sobre una muestra propuesta (x, r).

$$d_L(M_r, h(x)) = \sum_{s=1}^{l} L(M(r,s)h_s(x))$$

  donde $L(z)$ es la correspondiente *función de pérdida*. La etiqueta de clase predicha $y \in 1, ..., c$ se calcula análogamente al caso de la decodificación de Hamming. La *función de pérdida L* se adapta al algoritmo de aprendizaje. En el caso de este trabajo, la más acertada para C4.5, naive Bayes y kNN es la función de regresión logística, $L(z) = log(1 + e^{-2z})$, mientras que para el algoritmo SVM se utiliza la función $L(z) = (1 - z)_+$ [2].
- *Suma:* este método, para cada clasificador binario, va sumando las probabilidades asignadas a cada etiqueta. Una vez hecho el recuento de las probabilidades de todas las asignaciones realizadas por los clasificadores binarios, a cada patrón de prueba se le asignará la clase que tenga una mayor probabilidad acumulada.
- *Suma umbral:* este método es una modificación del anterior, que consiste en establecer un umbral según el cual solamente se suma la probabilidad asignada a una clase si esta es superior a dicho umbral. Además, cuando una clase fue asignada con una probabilidad mayor que el umbral, esa probabilidad es asignada a la clase seleccionada, pero al complementario ya no se le suma la probabilidad acumulada de la otra clase. Nótese que estos dos últimos métodos de unión han sido diseñados ad-hoc por los autores [3].

3. Materiales y métodos

3.1. Conjuntos de datos

Aunque en el desarrollo inicial del análisis de microarrays era difícil encontrar conjuntos de datos para llevar a cabo su estudio, en los últimos años ha crecido el número

de repositorios públicos para la comunidad científica. Para este trabajo, se han seleccionado diez conjuntos de datos con múltiples clases, cuyas principales características se describen en la Tabla 1.

Tabla 1. Resumen de las principales características de los microarrays multiclase

	# Clases	# Caract.	# Muestras	Distribución	Disponible en
CLL-SUB-111	3	11340	111	11-49-51	[17]
Leukemia 1	3	5327	72	38-9-25	[18]
Leukemia 2	3	11225	72	28-24-20	[18]
Brain Tumor 2	4	10367	50	14-7-14-15	[18]
SRBCT	4	2308	83	29-25-11-18	[18]
TOX-171	4	5748	171	45-45-39-42	[17]
Brain Tumor 1	5	5920	90	60-10-10-4-6	[18]
Lung cancer	5	12600	203	139-17-21-20-6	[18]
9-Tumors	9	5726	60	9-7-8-6-6-8-8-2-6	[18]
11-Tumors	11	12533	174	27-8-26-23-12-11-7-26-6-14-14	[18]

En la tabla se pueden observar algunas de las problemáticas que presenta este tipo de conjuntos de datos. En primer lugar, el pequeño tamaño muestral en relación al número de características. El número de muestras en los conjuntos de datos seleccionados oscila entre 50 y 203, mientras que el número de características va de 2308 a 12600. Además, la mayoría de estos conjuntos están desbalanceados, especialmente Lung cancer, Brain Tumor 1, CLL-SUB-111 y 9-Tumors.

3.2. Medidas de complejidad

Cuando los clasificadores no logran obtener una precisión perfecta en aplicaciones reales, las posibles causas pueden ser deficiencias en los algoritmos de clasificación, dificultades intrínsecas de los datos y desajustes entre métodos y problemas. Para abordar esta cuestión, se utilizarán una serie de medidas propuestas por Ho y Basu [12] para analizar el comportamiento de los clasificadores más allá de las estimaciones de las tasas de error.

1. Medidas de solapamiento en atributos de clases diferentes:
   - Razón discriminante de Fisher (F1)
   - Volumen de la región de solapamiento (F2)
   - Máxima eficiencia individual de los atributos (F3)
2. Medidas de separabilidad de clases
   - Suma del error de la distancia minimizada por programación lineal (L1)
   - Error del clasificador lineal por programación lineal (L2)
   - Fracción de puntos en los bordes de las clases (N1)

- Media de la distancia de vecinos más cercanos intra/inter clases (N2)
- Error del clasificador 1-NN (N3)

3. Medidas de geometría, topología y densidad
   - No-linealidad del clasificador lineal por programación lineal (L3)
   - No-linealidad del clasificador 1-NN (N4)
   - Fracción de puntos con subconjuntos adheridos (T1)
   - Media de puntos por dimensión (T2)

3.3. Algoritmos de clasificación

Cuatro clasificadores, pertenecientes a diferentes familias, fueron elegidos para evaluar la complejidad de los conjuntos de datos. Obsérvese que dos de ellos son lineales (naive Bayes y SVM que usa un *kernel* lineal) mientras que los otros dos no lo son (C4.5 y kNN).

- El algoritmo **C4.5** fue desarrollado por Quinlan como una extensión del algoritmo ID3 [15] y, al igual que éste, está basado en árboles de decisión. Un árbol de decisión clasifica una muestra, filtrándola de manera descendente, hasta encontrarse con una hoja, que corresponde a la clasificación buscada.
- **Naive Bayes** (NB) es un clasificador [16] que emplea la regla de Bayes para determinar la probabilidad de pertenecer a cada clase para una determinada muestra. Este clasificador asume que los atributos son condicionalmente independientes entre sí dada la clase. A pesar de su diseño 'ingenuo' y su aparente sencillez, los clasificadores naive Bayes son eficientes y robustos ante el ruido y los atributos irrelevantes.
- **K-Nearest Neighbor** (k-NN) [1] es una estrategia de clasificación donde un objeto se clasifica de acuerdo al voto mayoritario de sus vecinos, con el fin de ser asignado a la clase más similar de entre los k vecinos más cercanos (donde k es una constante definida por el usuario, 1 para este trabajo).
- El algoritmo **Support Vector Machines** (SVM) es un tipo de clasificador de patrones basado en técnicas estadísticas de aprendizaje propuestas por Vapnik [19]. En muchas aplicaciones, la utilización del algoritmo SVM ha mostrado tener un gran rendimiento, en parte por permitir fronteras de decisión flexibles y también por su buena capacidad de generalización.

3.4. Métodos de selección de características

Los métodos de selección de características han recibido una atención especial en la literatura al ser considerados como un paso esencial previo a la clasificación cuando trabajamos con conjuntos de alta dimensión. Generalmente, se dividen en tres tipos: *filtros*, *wrappers* y *métodos embebidos*. En este trabajo utilizaremos dos métodos de filtrado ya que parecen los más adecuados para este tipo de datos, fundamentalmente por su rapidez y su independencia respecto al clasificador.

- El filtro **Correlation-Based Feature Selection** (CFS) [10] es un sencillo algoritmo multivariado cuya principal meta es obtener el subconjunto de características más correlacionado con la clase y menos correlacionado entre sí.

- El **filtro basado en la consistencia** (CONS) [4] evalúa el valor de un subconjunto de características por el nivel de consistencia en los valores de la clase cuando las muestras de entrenamiento son proyectadas sobre un subconjunto de características.

4. Experimentación

En esta sección se presentan los resultados experimentales sobre los diez conjuntos de datos microarray descritos en la Tabla 1. Primero, analizamos la complejidad teórica de los conjuntos seleccionados. En la segunda parte, compararemos las diferentes estrategias de clasificación en base a la precisión obtenida por cuatro clasificadores.

4.1. Complejidad

La mayoría de las medidas de complejidad citadas en la Sección 3.2 han sido diseñadas para problemas binarios. Una solución sencilla para poder aplicar estas medidas es transformar el problema original multiclase en diferentes subproblemas binarios. Para ello, en este trabajo, utilizaremos las estrategias *one-versus-one* y *one-versus-rest*.

En la Tabla 2 se muestra un breve resumen de los resultados obtenidos por las diferentes medidas de complejidad sobre los diez conjuntos de datos seleccionados. Para cada particularidad de los datos (solapamiento, no-linealidad, proximidad a la frontera que divide las clases y dispersión de los datos) se indicarán aquellas clases que la presenten, así como las medidas de complejidad que reflejan estas propiedades. Por ejemplo, en Brain Tumor 1, la clase 0 no sería linealmente separable del resto de clases del conjunto. Si todas las clases del conjunto de datos presentan una determinada problemática, se marcará dicha casilla con un '*'. Finalmente, en la última columna se muestra la relación de desbalanceo (IR), definida como el número de muestras de la clase mayoritaria dividida por el número de muestras de la clase minoritaria. Un valor alto de IR indica que el conjunto de datos está extremadamente desbalanceado.

Tabla 2. Resumen de la complejidad teórica de los conjuntos de datos seleccionados

	# Clases	Solapamiento (F1, F3)	No-linealidad (L1, L2)	Proximidad frontera (N1)	Dispersión (N2)	IR
CLL-SUB-111	3	1,2	1,2	1,2	*	4.63
Leukemia 1	3		0,2		*	4.22
Leukemia 2	3	1	0,1,2		*	1.40
Brain Tumor 2	4	2	0,2,3	2	*	2.14
SRBCT	4				*	2.63
TOX-171	4	*	*		*	1.15
Brain Tumor 1	5	0,4	0		*	15.00
Lung cancer	5	0	0		*	23.16
9-Tumors	9	0,1,2,3		1,5	*	4.50
11-Tumors	11	2,9			*	4.50

Según estas medidas, los conjuntos más complejos serían TOX-171, 9-Tumors, Brain Tumor 2 y CLL-SUB-111, mientras que SRBCT no debería de presentar demasiados problemas en la tarea de clasificación. Como podemos ver en la tabla, una problemática común a todos estos conjuntos es la dispersión de muestras pertenecientes a la misma clase en el espacio de características, hecho que dificultaría la tarea de clasificación de algoritmos basados en distancias, como kNN.

4.2. Clasificación

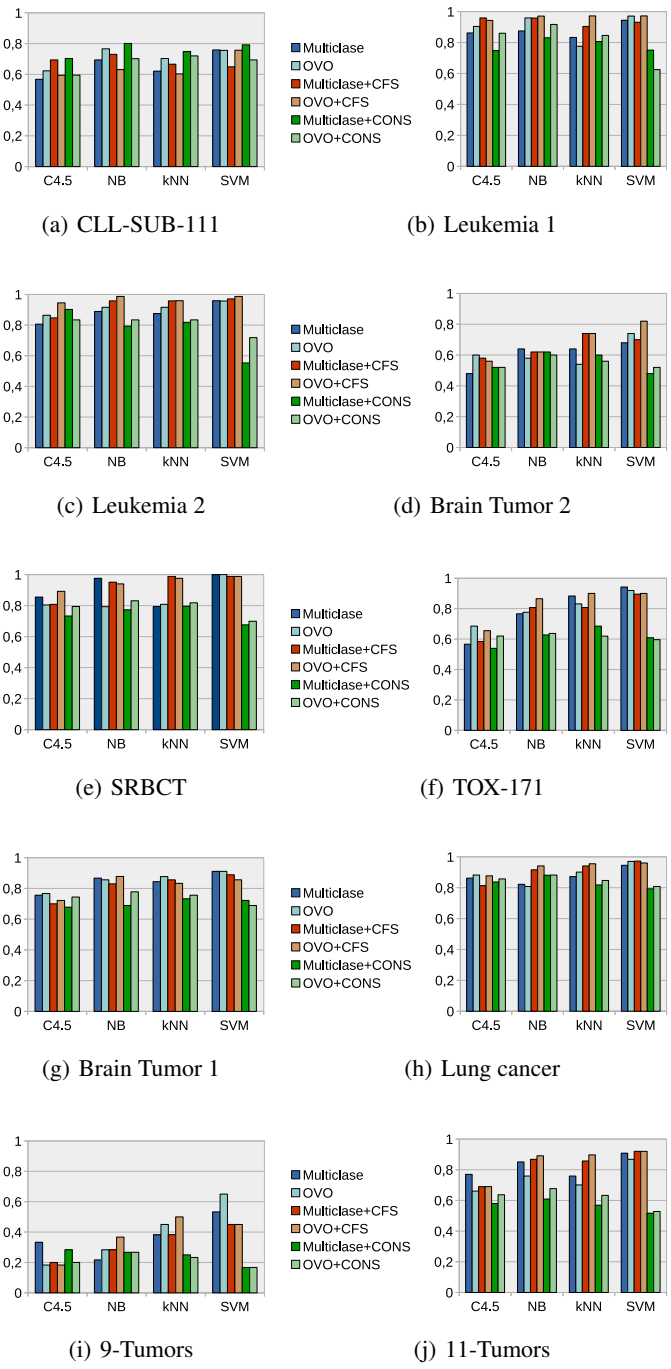
En la Figura 2 se puede observar la precisión de clasificación máxima obtenida por cada una de las estrategias para los conjuntos de datos seleccionados tras realizar una validación cruzada con 5 paquetes. Recordemos que para la segunda aproximación, donde empleamos el esquema OVO, los resultados proporcionados por cada clasificador se combinan utilizando cuatro técnicas de decodificación, aunque solamente se muestran en este artículo los resultados obtenidos por la mejor de ellas.

Antes de comenzar a analizar los resultados, es importante tener en cuenta que el algoritmo SVM es de naturaleza binaria, por lo que en este caso compararemos la precisión de clasificación obtenida por la implementación de este algoritmo en la herramienta Weka [9] para problemas con múltiples clases y nuestra versión OVO con los diferentes métodos de decodificación. En la implementación en Weka de SVM, los problemas con múltiples clases se resuelven utilizando la *clasificación pairwise*, que consiste en la aplicación de la estrategia OVO y la utilización de modelos logísticos [11].

Es lógico suponer que la estrategia OVO obtendrá mejores resultados que la aproximación multiclase en aquellos conjuntos que se encuentren más desbalanceados, especialmente en los que una clase presente un número de muestras significativamente mayor que el resto de clases. Esta situación se produce en los conjuntos Brain Tumor 1 y Lung cancer, los cuales tienen una clase que representa el 66.67 % y el 68.47 % del total de muestras del conjunto, respectivamente. En estos casos, la aproximación mediante el esquema OVO consigue la mejor precisión de clasificación para todos los clasificadores excepto NB.

El problema del desbalanceo de clases es bastante común en microarrays. Sin embargo, entre los conjuntos que hemos seleccionado hay dos cuyas clases presentan una distribución de muestras bastante similar. Nos referimos a Leukemia 2 y TOX-171. A pesar de ello, no vemos una superioridad clara del clasificador multiclase frente al esquema OVO. De hecho, el conjunto de datos Leukemia 2 obtiene los mejores resultados de clasificación gracias a esta última aproximación. Esto puede ser debido a los problemas de complejidad que presentan algunas de las clases de estos conjuntos (ver Tabla 2). También merecen una mención especial los conjuntos 11-Tumors y SRBCT, los cuales no presentan en principio problemas de complejidad ni sus muestras están demasiado desbalanceadas. En este caso, se observa que el clasificador multiclase obtiene la mejor precisión de clasificación independientemente del algoritmo empleado.

En cuanto a la aplicación de los métodos de selección de características, nuestros experimentos mantienen la hipótesis de que la mayoría de los genes no son relevantes para una clasificación precisa. La aplicación del filtro CFS consigue mejorar el rendimiento de clasificación en prácticamente todos los conjuntos, independientemente del algoritmo de clasificación empleado.



(a) CLL-SUB-111 (b) Leukemia 1

(c) Leukemia 2 (d) Brain Tumor 2

(e) SRBCT (f) TOX-171

(g) Brain Tumor 1 (h) Lung cancer

(i) 9-Tumors (j) 11-Tumors

Figura 2. Precisión de clasificación tras realizar validación cruzada con 5 paquetes

5. Conclusiones

Este trabajo tenía como principal objetivo realizar un análisis comparativo entre diferentes aproximaciones para clasificar problemas que involucran más de dos clases. La primera aproximación consistía en la aplicación directa de un clasificador para múltiples clases. La segunda, la transformación del problema original multiclase en varios problemas de clasificación binaria, de forma que la adquisición de conocimiento se realizaría desde los clasificadores binarios especializados. Para llevar a cabo las pruebas experimentales correspondientes se han utilizado diez conjuntos de datos microarray, con un número de clases entre tres y once. Además, debido a la alta dimensión de este tipo de conjuntos, se aplicaron dos métodos de selección de características.

Los resultados experimentales obtenidos por las diferentes estrategias de clasificación muestran una clara superioridad del esquema OVO frente al clasificador multiclase, especialmente después de aplicar los métodos de selección de características. De hecho, OVO está presente en ocho de las diez estrategias de clasificación ganadoras. Características propias de este tipo de conjuntos como el solapamiento entre clases, la no-linealidad de los datos o el desbalanceo de clases pueden influir en esta superioridad.

Por último, y haciendo referencia a las medidas de complejidad y las dos estrategias seguidas para dividir el problema original multiclase, el esquema OVR debería ser utilizado en este caso, especialmente cuando aumenta el número de clases del conjunto. Este esquema no solo es más sencillo de implementar sino que facilita la comprensión de los resultados, realizando una conexión más directa entre la complejidad teórica de una determinada clase y su tasa de verdaderos positivos.

A la vista de las conclusiones obtenidas, sugerimos como futura línea de investigación la aplicación de técnicas de *oversampling* para tratar de corregir el problema del desbalanceo de clases y estudiar cómo repercutiría este hecho en los resultados de clasificación obtenidos por las diferentes estrategias propuestas en este trabajo.

6. Agradecimientos

Esta investigación ha sido respaldada en parte por el Ministerio de Economía y Competitividad del Gobierno de España a través del proyecto de investigación TIN 2012-37954, parcialmente financiada por los fondos FEDER de la Unión Europea; y por la Consellería de Industria de la Xunta de Galicia a través del proyecto de investigación GRC2014/035. V. Bolón-Canedo agradece el apoyo de la Xunta de Galicia bajo el código de la beca postdoctoral ED481B 2014/164-0.

Referencias

1. D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, (1):37–66, 1991.
2. Erin L Allwein, Robert E Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *The Journal of Machine Learning Research*, 1:113–141, 2001.

3. V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos. Feature selection and classification in multiple class datasets: An application to kdd cup 99 dataset. *Expert Systems with Applications*, 38(5):5947–5957, 2011.
4. M. Dash and H. Liu. Consistency-based search in feature selection. *Artificial intelligence*, 151(1):155–176, 2003.
5. T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.
6. G. Forman. An extensive empirical study of feature selection metrics for next classification. *The Journal of Machine Learning Research*, 3:1289–1305, 2003.
7. T.R. Golub, D. K. Stomin, and P.Tamayo et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, 286(5439):531–537, 1999.
8. I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.
9. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.
10. M. A. Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.
11. Trevor Hastie and Robert Tibshirani. Classification by pairwise coupling. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. MIT Press, 1998.
12. T. K. Ho and M. Basu. *Data complexity in pattern recognition*. Springer, 2006.
13. G. Holmes, B. Pfahringer, R. Kirkby, E. Frank, and M. Hall. Multiclass alternating decision trees. *Proceedings of the European Conference on Machine Learning (ECML'02)*, 2430:161–172, 2002.
14. T. Li, C. Zhang, and M. Ogihara. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, 20(15):2429–2437, 2004.
15. J. R. Quinlan. C4.5: programs for machine learning. Morgan Kaufmann Publishers, 1993.
16. I. Rish. An empirical study of the naive Bayes classifier. *in IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, (22):41–46, 2001.
17. Arizona State University. Feature selection datasets. Disponible en: `http://featureselection.asu.edu/datasets.php` [Online; Último acceso: Septiembre-2015].
18. Vanderbilt University. Gene expression model selector. Disponible en: `http://www.gems-system.org/` [Online; Último acceso: Septiembre-2015].
19. V. N. Vapnik. *Statistical learning theory*. Wiley, 1998.
20. N. Yukinawa, S. Oba, K. Kato, and S. Ishii. Optimal aggregation of binary classifiers for multi-class cancer diagnosis using gene expression profiles. *IEEE/ACM Transactions on Computacional Biology and Bioinformatics*, 6(2):333–343, 2007.