Selección de características

Paloma Pérez G.
paperez@dcc.uchile.cl
Inteligencia artificial, EL4106 - Tarea 4

Resumen

En el presente trabajo se exponen los efectos de la selección de atributos sobre dos conjuntos de datos diferentes: datos médicos relacionados con la determinación de diabetes en pacientes y de datos relacionados con escritura a mano alzada de dígitos (los cuales son analizados con un software destinado a la lectura de documentos manuscritos). El clasificador usado para ambos conjuntos fue el de SMO, usando una aproximación lineal. El primer conjunto mencionado (diabetes) tuvo un comportamiento positivo al momento de seleccionar atributos, y por tanto disminuyendo en número. Por el contrario, el conjunto de datos de dígitos se vio afectado negativamente con la disminución de sus dimensiones con la elección de características.

Keywords – características, selección, entrenamiento, prueba

1. Introducción

Tanto problemas de estadística como de inteligencia artificial es posible encontrar dificultades al momento de determinar que información es útil o relevante para poder extraer conclusiones más precisas respecto del fenómeno que se esté estudiando. Esta información relevante o útil no es simple de detectar al observar los datos debido a que estos podrían tener una gran dimensionalidad o su influencia en los resultados no es del todo obvia. Para estos casos, en los cuales se busca construir un modelo de clasificacón sencillo pero a su vez lo más calibrado posible se debe explorar la información entregada a este y determinar que información es la más apropiada para tal acometido. Para este fin, es necesario hacer una selección de la información la que se traduce en la selección de características o atributos de las instancias de las clases que se quiere detectar.

En este trabajo se pretende realizar un acercamiento al proceso de selección de atributos usando dos diferentes tipos de bases de datos: una relacionada con la determinación de diabetes y la otra relacionada con la determinación de dígitos a partir de información obtenida de manuscritos. Este acercamiento se realizará usando el software WEKA.

1.1. Estructura del informe

Las siguientes secciones corresponden a: **Marco teórico**, en el cual se responde a preguntas planteadas respecto del la selección de características, de su proceso y que tipo métodos podrían apoyar en el proceso. Posteriormente, en la sección 3 se estudian los resultados obtenidos para las diferentes experiencias con el conjunto de datos de diabetes. En la sección 4 se describe la transformación del archivo CSV de los datos de dígitos a un archivo ARFF (para poder cargarlo en el software WEKA), y de la clasificación sin y con selección de atributos para estos datos. Luego en la sección 5 se tiene un análsis global de los resultados más importantes, que luego son resumidos en la sección de conclusiones 6.

2. Marco teórico

2.1. Selección de características

En aprendizaje de máquinas y estadística, la *selección de características* corresponde esencialmente a la selección de un subconjunto de atributos relevantes para la construcción de un modelo. Esta selección se realiza por las siguientes razones:

1. Simplificación del modelo para lograr una interpretación más fácil de conseguir.
2. Disminuir los tiempos de entrenamiento
3. Para evitar la **maldición de la dimensionalidad** [1], la cual no sólo complejiza el entrenamiento, sino también exige una mayor cantidad de datos de entrenamiento.
4. Lograr una mejor generalización del modelo, reduciendo el overfitting (i.e. para reducir la varianza).

Su finalidad es la de deshacerse de información irrelevante o que no aporta información significativa dentro del modelo.

2.2. Etapas de la selección de características

1. **Generación de subconjuntos:** Es un procedimiento de sondeo que genera subconjuntos candidatos de características para la evaluación, basándose en una *estrategia de búsqueda* [2].
2. **Evaluación:** Cada subconjunto candidato es evaluado y comparado con los resultados obtenidos por el mejor subconjunto previo (el que ha obtenido mejores resultados en clasificación) de acuerdo a un *criterio de evaluación*. La generación de estos conjuntos dependerá de la métrica de evaluación, la cual puede corresponder a una de las siguientes categorías: de filtrado, wrappers o método embebido.
3. **Criterio de parada:** El criterio de parada determina cuando el proceso de selección de características debiese de detenerse. Algunos criterios de detención usados corresponden a: un número suficiente de buenos subconjuntos es seleccionado, la agregación de nuevas características ya no produce una mejora en el subconjunto, alguna condición límite es alcanzada como el número mínimo o máximo

de características o de iteraciones en la generación de subconjuntos, o simplemente la finalización de la búsqueda de subconjuntos (no hay más subconjuntos por generar).

4. **Validación:** El subconjunto seleccionado usualmente requiere de ser validado a través de conocimiento que se tenga de los datos o a través de diferentes pruebas usando datos sintéticos o reales.

2.3. Metodologías para la generación de subconjuntos

1. **Completa:** Garantiza la obtención del resultado óptimo de acuerdo al criterio de evaluación usado . Mientras que una búsqueda exhaustiva es completa, una búsqueda no requiere ser exhaustiva para asegurar completitud. Algunos ejemplos de métodos completos son *branch and bound* y *beam search*.

2. **Heurística:** La generación de subconjuntos es esencialmente una búsqueda heurística, cuyo espacio de estados está definido por la cantidad de subconjuntos a ser evaluados. Es deber del investigador determinar el subconjunto de partida y que dirección tomar (agregar y/o quitar atributos) [2].

3. **Aleatoria:** Comienza con una selección aleatoria de algún subconjunto y luego procede en dos direcciones diferentes: una es realizando una búsqueda secuencial aleatoria (por ejemplo, *random-start hill-climbing*, *simulated annealing*), la otra dirección corresponde a generar el próximo subconjunto de forma completamente aleatoria (usando el algoritmo de *Las Vegas*).

2.4. Metodologías de evaluación de subconjuntos

1. **Basada en distancia:** Corresponden a las metodologías llamadas también de separabilidad, divergencia o de medidas de discriminación. Para el problema de dos clases se preferen los atributos que aporten la mayor distancia posible entre las clases.

2. **Basada en información:** Corresponde a la medida de la ganancia de información por atributo: la ganancia está definida por la diferencia entre la incerteza del prior y del posterior esperado de la clase usando un atributo específico. Un atributo es preferido respecto de otro si la ganancia de información es mayor.

3. **Basada en dependencia:** Relacionado con la medida de correlación o similitud. Se detecta la dependencia entre atributos, si alguno de estos es función de otro o puede ser predicho por tal. Un atributo es preferido respecto del otro si la asociación del primero con la clase que se quiere determinar es mayor que la asociación del segundo atributo con la misma clase.

4. **Basada en consistencia:** Estas metodologías dependen fuertemente en la información de la clase y del bias que genera el criterio de la mínima cantidad de atributos al momento de seleccionar subconjuntos de características. Esta metodología intenta encontrar el subconjunto de menor tamaño que pueda separar las clases consistentemente. Una inconsistencia se define como dos instancias que tienen los mismos valores en determinadas características pero son pertenecientes a diferentes clases.

5. **Basada en error de clasificación:**

3. Selección de características y clasificación de datos de diabetes

En esta sección usamos un clasificador SMO con los valores prestablecidos (o por defecto) definidos en el software WEKA, con el cual entrenamos el 66 % de los datos, mientras que el resto es destinado a la evalución (conjunto de prueba).

3.1. Clasificación por defecto

La clasificación obtenida usando todas las características de los datos de diabetes entrega la siguiente matriz de confusión.

	NEG	POS
NEG	161	17
POS	37	46

Cuadro 1: Matriz de confusión obtenida para datos de diabete usando todos los atributos.

De estos resultados se desprende que se logra un 79.31 % de instancias correctamente clasificadas.

3.2. Clasificación seleccionando características usando BestFirst y WrapperSubsetEval

Para este experimento usamos el seleccionador de conjuntos BestFirst y el evaluador WrapperSubsetEval. Este último debe ser configurado para poder realizar las evaluaciones de los subconjuntos usando SMO. En esta ocasión, los resultados obtenidos por la selección de características indican que la información aportada por el feature *skin* no es relevante. Descartando este atributo se obtiene la siguiete matriz de confusión.

	NEG	POS
NEG	160	18
POS	36	47

Cuadro 2: Matriz de confusión obtenida descartando el atributo *skin*.

De estos resultados se desprende que se logra un 79.31 % de instancias correctamente clasificadas. Es decir, se mantiene el porcentaje anterior.

3.3. Clasificación seleccionando sólo 4 características usando Ranker e InfoGainAttributeEval

Para este experimento usamos el seleccionador de conjuntos Ranker y el evaluador InfoGainAttributeEval. En esta oportunidad se estima que los atributos *plas*, *insu*, *mass* y *age* son las cuatro primeras características que mejor describen el modelo aportando información significativa.

Usando estas features se obtiene un 80.07 % de instancias correctamente clasificadas.

	NEG	POS
NEG	159	19
POS	33	50

Cuadro 3: Matriz de confusión resultante al usar los cuatro *mejores* atributos usando Ranker e InfoGainAttributeEval

3.4. Clasificación seleccionando sólo 2 características usando Ranker e InfoGainAttributeEval

En este experimento nuevamente usamos Ranker y el evaluador InfoGainAttributeEval, sin embargo esta vez escogemos las 2 mejores features, las que resultan ser *plass* y *mass*.

	NEG	POS
NEG	164	14
POS	36	47

Cuadro 4: Matriz de confusón usando los dos mejores atributos: *plas* y *mass*

En esta oportunidad, el resultado de instancias correctamente clasificadasa mejoró sustancialmente, alcanzando un 80.84 %.

4. Selección de atributos y clasificación de dígitos

4.1. Obtención de archivo ARFF

La transformación del CSV que contiene los datos de los dígitos manuscritos fue realizada en PYTHON, a través del siguiente código:

Listing 1: csv2arff.py

```
1 def write_arff(filepath):
2     data=np.loadtxt(filepath, delimiter=',',dtype=int)
3     rows, cols = data.shape
4     arff_head = write_str(cols-1)
5     np.savetxt('digits.arff', data, delimiter=',', fmt='%d',header=↩
        arff_head)
6
7 def write_str(n_attrs):
8
9     rel = '@relation orh_digits\n'
10    str_tmp = '@attribute \'attr_%d\' numeric\n'
11    str_attrs = ''
12
13    for i in range(1, n_attrs+1):
14        str_attrs = str_attrs +str_tmp  % i
15
16    str_cl = '@attribute \'class\' { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }\n'
17
18    return rel+str_attrs+str_cl+'@data\n'
19
20 if __name__ == "__main__":
21     write_arff('./1_digits.txt')
```

La función `write_str` recibe como parámetros la cantidad de atributos que se encontraron en la base de datos, y procede a la construcción de un *header* de archivo ARFF, a partir de esta cantidad, concatenando el string que etiqueta los atributos. Como estos no tienen nombre (originalmente, pueden ser identificados por so posición en la instancia) se les identificó como *attr_n* donde *n* es un número entero, y corresponde a la posición en la secuencia de carácterísticas, comenzando desde 1.

Con las clases definidas en un string, se procede a concatenar este string para retornarlo en `write_arff`. Este método recibe como entrada el archivo *1_digits.txt*, el cual se carga como matriz usando la librería **numpy**, para obtener la cantidad de caracteres y poseteriromente ser reescrita en en el nuevo archivo ARFF, denominado *digits.arff*, usando el header entregado por write_str.

4.2. Clasificación por defecto

La primera clasificación fue realizada usando un modelo SM, entrenandolo con un 66 % y evaluando con el resto de los datos. Este experimento se realizó considerando todos los atributos del dataset original (64 en total). Se logró clasificar correctamente

	0	1	2	3	4	5	6	7	8	9
0	178	0	0	0	0	0	0	0	0	0
1	0	176	0	0	1	0	0	0	1	2
2	0	0	189	1	0	0	1	0	0	0
3	0	0	0	198	0	0	0	0	0	1
4	0	0	0	0	203	0	0	0	0	0
5	0	0	0	0	0	190	0	0	0	0
6	0	0	0	0	2	0	216	0	0	0
7	0	0	0	1	0	0	0	171	0	0
8	0	2	0	1	1	1	1	1	177	2
9	0	0	0	2	1	0	0	0	1	190

Cuadro 5: Matriz de confusión obtenida para datos de dígitos usando los 64 atributos.

alrededor de 98.8 % instancias.

4.3. Clasificación seleccionando características con BestFirst y WrapperSubsetEval

Este experimento consideró la selección de atributos usando como generador de subconjuntos el método BestFirst y evaluador el mátodo WrapperSubsetEval. Esta parte se realizó en la pestaña de subprocess de WEKA, el cual finalmente consideró sólo 39 atributos relevantes.

	0	1	2	3	4	5	6	7	8	9
0	178	0	0	0	0	0	0	0	0	0
1	0	175	1	0	0	0	0	0	2	2
2	1	0	190	0	0	0	0	0	0	0
3	0	0	0	197	0	1	0	0	0	1
4	1	1	0	0	199	0	0	0	0	2
5	1	0	0	0	0	189	0	0	0	0
6	0	0	0	0	3	0	215	0	0	0
7	1	0	0	2	0	0	0	169	0	0
8	0	8	1	1	1	1	0	1	171	2
9	0	0	0	2	0	0	0	0	1	191

Cuadro 6: Matriz de confusión obtenida en la clasificación de dígitos usando 39 atributos.

Esta nueva clasificación (usando igualmente un modelo SMO) consiguió un porcentaje de instancias correctamente clasificadas de alrededor de 98.06 %, algo menor al porcentaje obtenido usando todas las características.

5. Análisis

5.1. Clasificación datos diabetes

La tabla 7 muestra la relación, para el conjunto de datos de diabetes, que la selección de atributos mejora la clasificación para el modelo SMO. Además al ocupar menos características, usando un rankeo de los atributos, se observa una notable mejora en los resultados.

NC	ICC [%]
8	79.31
7	79.31
4	80.07
2	80.8

Cuadro 7: Número de características (NC) versus porcentaje de instancias correctamente clasificadas (ICC)

5.2. Clasificación datos dígitos

La tabla 8 muestra el efecto de la selección de atributos usando BestFirst y WrapperSubsetEval en el conjunto de datos de los dígitos. En esta oportunidad, la selección sólo empeora los resultados en la clasificación.

NC	ICC [%]
64	98.8
39	98.06

Cuadro 8: Número de características (NC) versus porcentaje de instancias correctamente clasificadas (ICC)

5.3. Reducción de características

En ambos conjuntos de datos se observan efectos diferentes en la selección de atributos: mientras que las instancias de diabetes mantiene 2 o mejora su clasificación 34, el conjunto de datos de dígitos empeora sus resultados al pasar de 64 características a 39.

6. Conclusión

La mejor solución para el dataset de diabetes correspondió a la selección de las cuatro mejores características usando Ranker como método de generación de subconjuntos e InforGainAttributeEval como método evaluador, para el modelo SMO. Por el contrario, el conjunto de datos de dígitos se vio afectado negativamente por la selección de atributos, lo que indicaría que, una reducción en la información entregada al modelo de entrenamiento puede perjudicar el proceso de clasificación. Esto último puede deberse a la subestimación por parte del método evaluador como WrapperSubsetEval o una mala selección del método de generación de conjuntos como BestFirst para SMO en particular (en esta tarea sólo se empleó el modelo SMO).

Cabe destacar, como se mencionó anteriormente para el caso de los datos de diabetes, la clasificación mejoró sustancialmente al disminuir la cantidad de características a menos de la mitad. Esto podría insinuar que hay datos cuya clasificacón es más o menos sensible a la selección de atributos y puede ser afectada de manera negativa (clasificación de dígitos) o positiva (clasificación de diabetes). Esta sensibilidad estará determinada por la dependencia del modelo respecto de los atributos (a veces, dos o más atributos por si solos no aportan información relevante, pero juntos si lo hacen).

Es probable que los resultados cambien para ambos datasets usando modelos de clasificadores diferentes (por ejemplo, un modelo no-lineal para la clasificación de los datos de los dígitos).

Referencias

[1] R. Bellman, *Dynamic programming*. Courier Corporation, 2013.

[2] H. Liu and L. Yu, "Yu, l.: Toward integrating feature selection algorithm for classification and clustering. ieee transaction on knowledge and data engineering 17(4), 491-502," vol. 17, pp. 491–502, 04 2005.