Clasificación de Bosques por Información Cartográfica

Reconocimiento de Patrones 2015

Proyecto Estándar

Yamil Abraham - Mauricio González

Abstract

En este trabajo se plantea una solución al problema propuesto en el curso acerca de la clasificación de bosques. Aquí mostraremos la progresión en el procesado de los datos y cómo se llegó al clasificador escogido. Se utilizaron en forma paralela los programas Weka y MatLab; el primero se utilizó para observar tanto el espacio de características así como también distintos métodos convencionales para visualizar el carácter del problema. Una vez seleccionado el método (en nuestro caso árboles de decisión) se pasó a trabajar en MatLab para desarrollar un algoritmo en forma automática para el proceso de datos.

Por otra parte, se enfatizó el trabajo sobre el espacio de características. En esta instancia nos encontramos con inconvenientes como el solapamiento de clases y combinaciones de características con y sin métrica entre los datos. Finalmente se escogió desglosar el espacio de características entre aquellas que poseían métrica y aquellas que no. También se optó por trabajar las clases que más se confundían como si fuera una sola, y luego clasificarlas por separado.

Contents

1 Acerca del problema

1.1 Descripción

El problema a abordar consiste en clasificar muestras que corresponden a porciones de un bosque nativo del Estado de Colorado, EEUU. Las muestras están contenidas en un espacio de 54 dimensiones, y se clasifican de acuerdo a 7 posibles etiquetas.

El espacio de características puede dividirse en dos subespacios con comportamientos diferentes

- 10 características con valores escalares. Estas características representan información independiente, y a priori no podemos asumir que tienen correlación.
- 44 características con valores binarios, de las que 40 corresponden al tipo de suelo presente en el área estudiada y 4 al tipo de vegetación circundante.

Las características binarias presentan la problemática de que representan un determinado dato en un formato multidimensional, por lo que se hace difícil establecer una métrica que tenga sentido.

El set de muestras de entrenamiento está compuesto por 7612 patrones, divididos en 7 clases distintas.

1.2 Objetivo

Se debe proponer un clasificador que determine el tipo de árbol con un porcentaje de acierto no menor al 82%. El clasificador será determinado a partir del conjunto de entrenamiento, para después ser evaluado por el conjunto de test.

Los objetivos propuestos por el curso son los siguientes:

- Que el sistema tenga un porcentaje de acierto de al menos el 82% evaluado con el conjunto de datos de entrenamiento disponible. Se debe explicar la metodología seguida para mostrar que se cumple dicho desempeño.
- Explorar una forma de mejorar la información del tipo de suelo, agregando y/o sustituyendo las características existentes por otras. Evaluar si esto tiene impacto en el desempeño.
- Explorar el impacto que tiene aplicar alguna técnica de edición sobre los datos.
- Estimar el desempeño esperado para el conjunto de test, desconocido a priori.

1.3 Enfoque

El trabajo ralizado se puede dividir en tres etapas

- Explorar distintas técnicas de clasificación convencionales, evaluando detalladamente su efectividad para separar las muestras. Además del porcentaje de aciertos resulta interesante estudiar las diferentes matrices de confusión, ya que nos brindan información sobre qué clases son frecuentemente confundidas entre sí, lo que permite encontrar algoritmos complementarios.

- Estudiar el impacto de preprocesar los datos. En particular se estudiarán formas de tratar las clases binarias, ya que su formato multidimensional es poco amigable para muchas de las técnicas de clasificación que se pueden utilizar.

- Implementar la combinación de técnicas que se supone más efectiva en Matlab, ajustando sus parámetros para obtener los mejores resultados posibles.

1.4 Alcance

- Debe garantizarse el mínimo de elementos correctamente etiquetados impuesto por la propuesta inicial.

- Se evaluarán las técnicas de procesamiento y clasificación estudiadas en el curso, y se justificará el uso de las herramientas que sean seleccionadas para el procesamiento final.

- Se estudiarán distintas técnicas de preprocesamiento para los patrones del conjunto de entrenamiento, prestando especial atención a las características binarias.

- Se evluará la efectividad del algoritmo contra un conjunto de test.

2 Espacio de características y algoritmos utilizados

2.1 Análisis del espacio de características

En primera instancia se analizó el espacio de características en Weka para identificar dimensiones que no introdujeran información al sistema. Haciendo esto se descubrió que las características correspondientes a los valores de suelo 7, 8 y 19 nunca tomaban valor positivo, es decir, no hay ninguna muestra identificada con dicho sustrato. La existencia de estas dimensiones causa que las características se concentren en hiperplanos dentro del espacio en el que existen, lo cual es contraproducente para los algoritmos de clasificación.

Una vez eliminadas las características que no introducen información, se pasó a evaluar la conveniencia de eliminar las que agruparan muy pocas muestras. Para los casos en los que la información introducida es muy pobre es necesario estudiar el trade-off entre dicha información y la inclusión de una dimensión extra. El criterio adoptado fue eliminar las características que agruparan menos de 100 muestras.

Por otra parte, se planteó la posibilidad de discriminar las características binarias pertinentes estudiando su entropíá, pero se llegó a la conclusión de que dicho criterio resultaba redundante con los criterios de podado de los árboles de decisión que se describen en la sección 2.2.

Finalmente se evaluó la posibilidad de utilizar un algoritmo de selección de direcciones principales, pero ésto introduciría un concepto de métrica interclase que no tiene sentido para este caso particular. Esto se debe al método particular en el que se entregan los datos de tipo de suelo, que no resultan para nada convenientes para las técnicas de procesamiento que conocemos.

2.2 Algoritmos y línea de trabajo

Antes de modificar el espacio de características, se probaron clasificadores básicos en la plataforma Weka para así identificar aquellos que caracterizaran mejor al sistema. Los algoritmos testeados fueron NaiveBayes (NB), Support-Vector-Machine (SMO), árbol de decisión C4.5 (J48), Random Tree (RT) y Random Forest (RF). En la siguiente tabla se muestran los primeros resultados para estos clasificadores:

Clasificador	Patrones correctamente clasificados
NB	65,96 %
SMO	69,34 %
J48	77,26 %
RT	70,85 %
RF	83,70 %

Es evidente la diferencia que presenta el clasificador Random Forest en comparación al resto, por lo que se decidió trabajar con árboles de decisión para realizar los objetivos del proyecto. Sin embargo se testearon los mismos clasificadores junto con Adaboost, cuyos resultados fueron

Clasificador	Patrones correctamente clasificados
A.NB	65,96 %
A.SMO	69,34 %
A.J48	82,57 %
A.RT	70,98 %
A.RF	83,87 %

Llegado este punto, se pasó a utilizar únicamente árboles de decisión. Resta implementar un preprocesamiento adecuado en el espacio de características para optimizar los resultados.

Todos los clasificadores implementados presentaron errores similares: hay una gran confusión entre las clases 1 y 2, y entre las 3 y 6. Por lo tanto será conveniente trabajarlas como una única clase, para luego clasificarlas por separado. En la figura I se muestra la matriz de confusión para el clasificador Random Forest.

Antes de finalizar el trabajo con Weka se implementó un filtro *AttributeSelection.InfoGainAttributeEval*, el cual genera un ranking de características colocándolas en orden según la ganancia de información que proporcionan al sistema. Se seleccionaron subconjuntos de características con las primeras del ranking y se utilizó el clasificador *AdaboostM1.RandomForest* para los datos. En la tabla siguiente se muestran los resultados logrados para los distintos conjuntos de características.

Cantidad de características	A.RF
10	82,11 %
15	83,59 %
20	83,47 %
25	83,46 %
30	83,72 %
35	84,34 %

Vemos que hay una mejora en los datos al manipular el espacio de características. En este punto se pasó a trabajar en MatLab para tener un contacto más directo al preprocesar los datos.

```
=== Confusion Matrix ===

   a    b    c    d    e    f    g   <-- classified as
 844  128    1    0   40    4   89 |    a = 1
 205  525   28    0  102   41    9 |    b = 2
   0    4  890   69   15  163    0 |    c = 3
   0    0   14 1104    0   12    0 |    d = 4
   1   29   13    0 1030   14    0 |    e = 5
   0    7  137   38    6  965    0 |    f = 6
  45    1    0    0    1    0 1038 |    g = 7
```

Figura I - Matriz de confusión para Random Forest.

En MatLab se implementó un programa generador de árboles de decisión, basándose en la impureza de Gini como criterio de crecimiento. Esta impureza consiste en una medida de cuantas veces un elemento seleccionado arbitrariamente del conjunto sería etiquetado de forma incorrecta, habiendo sido etiquetado previamente de forma aleatoria. En forma analítica, la impureza de Gini de un conjunto puede calcularse como

$$I_G = \sum_{i=1}^{m} P_i \left(1 - P_i\right) = 1 - \sum_{i=1}^{m} P_i^2$$

donde se asume que el problema tiene m clases y que la probabilidad de un elemento de la clase i-ésima es P_i. El árbol de decisión entonces crece minimizando la impureza de Gini.

Sin alterar las características de los datos, y utilizando distintos valores de umbral de pospodado para χ_0^2, se obtuvo como resultado

Árbol de decisión	$\chi_0^2 = 10$	$\chi_0^2 = 5$	$\chi_0^2 = 1.5$
Porcentaje de aciertos	78,3368 %	85,6148 %	97,5828 %
Cantidad de patrones mal clasificados	1649	1095	184
Cantidad de nodos del arbol	319	829	2113

Esto pareciera ser el clasificador definitivo a utilizar con $\chi_0^2 = 1.5$, sin embargo al haber trabajado previamente en Weka es muy probable que haya un sobreajuste del clasificador. Otra evidencia de esto es la cantidad de nodos que posee el árbol, por lo que conluimos que su complejidad es inadecuada para nuestro problema.

Sabemos que hay clases que se confunden (1 y 2, 3 y 6) por lo tanto será conveniente generar un árbol de decisión unificando estas clases, para luego generar otros dos árboles que las trabajen por separado. Haciendo esto, nuestro problema de clasificación se convierte en tres problemas:

- Un problema de clasificación de 5 clases con 54 características.

- Dos problemas de clasificación de 2 clases con 54 características.

Repitiendo el procesamiento anterior para el sistema planteado, se obtuvo

5 clases	$\chi_0^2 = 10$	$\chi_0^2 = 5$	$\chi_0^2 = 1.5$
Porcentaje de aciertos	88,5181 %	93,4971 %	98,8833 %
Cantidad de patrones mal clasificados	874	495	85
Cantidad de nodos del arbol	255	599	1249

2 clases (1 y 2)	$\chi_0^2 = 10$	$\chi_0^2 = 5$	$\chi_0^2 = 1.5$
Porcentaje de aciertos	75,6448 %	80,2579 %	97,4206 %
Cantidad de patrones mal clasificados	491	398	52
Cantidad de nodos del arbol	29	85	555

2 clases (3 y 6)	$\chi_0^2 = 10$	$\chi_0^2 = 5$	$\chi_0^2 = 1.5$
Porcentaje de aciertos	76,4603 %	87,4455 %	97,9512 %
Cantidad de patrones mal clasificados	540	288	47
Cantidad de nodos del arbol	57	205	521

Se ve claramente que trabajar con las clases unificadas tiene un gran efecto en la clasificación de los datos. Se pasó de tener un único árbol de 2113 nodos a tres árboles de 1249, 555 y 521 nodos (en el caso $\chi_0^2 = 1.5$).

Finalmente, se buscó dar un último paso con el espacio de características utilizando PCA y un criterio de eliminación de características binarias. Dado que el espacio de características posee 44 dimensiones de carácter binario, habría que determinar en forma previa una métrica adecuada para poder aplicar el análisis de componentes principales. Esto es un problema excesivamente complejo, por lo que se procedió con el análisis únicamente en aquellas que poseían métrica de antemano. Aplicando el algoritmo, reducimos el espacio normado de 10 características a 6. En la figura II vemos el gráfico de los patrones para tres de las características principales halladas. Por otra parte, el criterio de eliminación de características de suelo que se utilizó consiste en lo siguiente:

- Calcular cuantos patrones poseen un tipo de suelo.
- Repetir para todos los suelos.
- Eliminar aquellos que tengan menos de n patrones asociados.

La figura III muestra el proceso mencionado en un esquema lógico. En este trabajo se utilizó $n = 100$, que representa el 1.31% del total de los patrones.

Aplicando ahora el clasificador al conjunto modificado, tenemos

5 clases	$\chi_0^2 = 10$	$\chi_0^2 = 5$	$\chi_0^2 = 1.5$
Porcentaje de aciertos	88,4919 %	93,4577 %	98,8439 %
Cantidad de patrones mal clasificados	876	498	88
Cantidad de nodos del arbol	361	697	1363

2 clases (1 y 2)	$\chi_0^2 = 10$	$\chi_0^2 = 5$	$\chi_0^2 = 1.5$
Porcentaje de aciertos	73,3631 %	81,9940 %	97,6190 %
Cantidad de patrones mal clasificados	537	363	48
Cantidad de nodos del arbol	11	129	543

2 clases (3 y 6)	$\chi_0^2 = 10$	$\chi_0^2 = 5$	$\chi_0^2 = 1.5$
Porcentaje de aciertos	74,4115 %	83,0863 %	98,0820 %
Cantidad de patrones mal clasificados	587	388	44
Cantidad de nodos del arbol	37	135	551

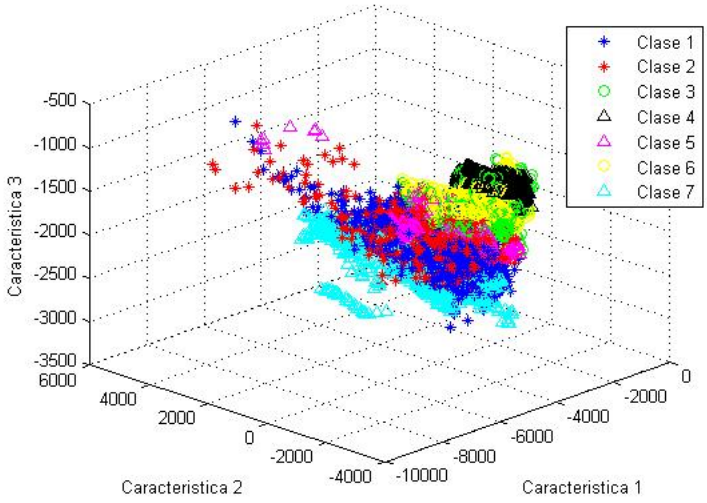

Figura II - Patrones en un subespacio tridimensional de direcciones principales.

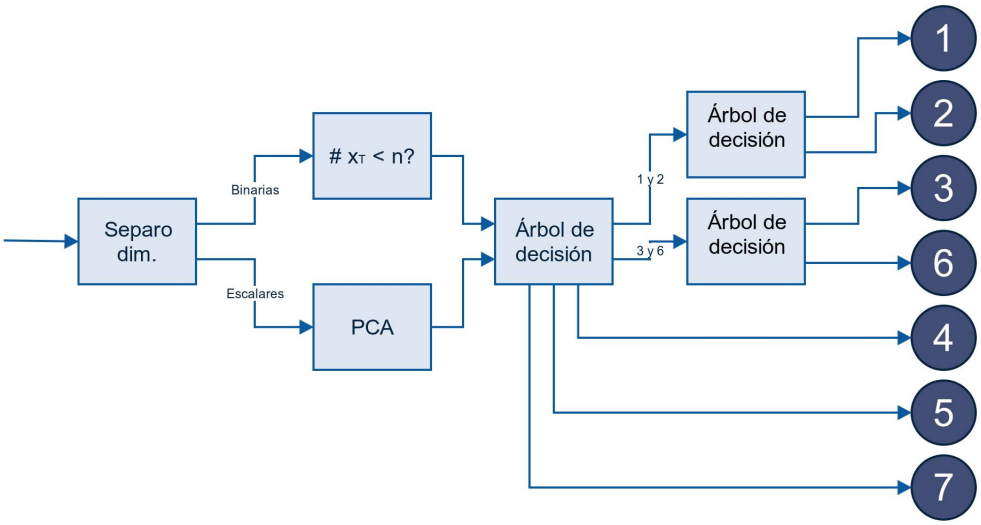

Figura III - Diagrama lógico de preprocesamiento sobre los datos.

3 Resultados y discusiones

Defiinido el algoritmo a utilizar, se probó su desempeño sobre el conjunto de test. Los resultados para $\chi_0^2 = 1.5$ fueron los siguientes:

5 clases	Total de los datos
Porcentaje de aciertos	87,6068 %
Cantidad de patrones mal clasificados	874
Cantidad de nodos del arbol	1363

2 clases	Clases 1 y 2	Clases 3 y 6
Porcentaje de aciertos	72,9295 %	80,4369 %
Cantidad de patrones mal clasificados	545	394
Cantidad de nodos del arbol	543	551

Considerando el caso $\chi^2 = 0.5$, el resultado fue

Datos con procesado completo	Total de los datos
Porcentaje de aciertos	87,8990 %
Cantidad de patrones mal clasificados	853
Cantidad de nodos del arbol	1545

Datos con procesado completo	Clases 1 y 2	Clases 3 y 6
Porcentaje de aciertos	73,2870 %	81,6286 %
Cantidad de patrones mal clasificados	538	370
Cantidad de nodos del arbol	645	643

En primer lugar podemos concluir que los resultados obtenidos son muy satisfactorios ya que superan ampliamente los mínimos exigidos. Comprobamos la efectividad de los árboles de decisión para resolver problemas de clasificación multiclase, y pudimos observar la conveniencia de utilizar algoritmos de boosting como Adaboost. También fue posible experimentar con combinaciones de árboles, verificando que para determinados subconjuntos problemáticos de clases mejora notoriamente los resultados. Durante este proyecto nos enfrentamos a problemas con el formato de los datos de entrada, algo que no se había experimentado en el curso hasta ahora. Se desarrollaron varios métodos para solucinar este problema, adquiriendo valiosos conocimientos sobre la importancia de los espacios de dimensiones

References

[1] Duda, Hart and Stork; Pattern Classification, John Wiley & Sons (ISBN-10-0471056693)-2001

[2] Bishop, C. M; Pattern Recognition and Machine Learning, Springer (ISBN-13- 9780387310732)-2006