

Textones y Clasificadores

Visión por Computador, Universidad de los Andes, Bogotá, Colombia

March 31, 2016

María Alejandra Bravo Sarmiento
ma.bravo641@uniandes.edu.co

Lina María Mejía López
lm.mejia11@uniandes.edu.co

Abstract

Para la clasificación de imágenes es necesario construir representaciones que nos permitan extraer las características más relevantes de las mismas. Una de las representaciones más útiles utilizadas son los diccionarios de textones. En este informe se presenta una aplicación del método de representación por medio de bag of words (BOW) a un problema de clasificación de texturas. Se utilizan dos clasificadores vecinos más cercanos (KNN) y bosques aleatorios (RF), estos se optimizan a partir de varios experimentos en los que se varían los parámetros propios de cada uno. Para KNN el mejor método obtuvo un resultado de ACA de 0.6400 y RF el mejor resultado obtenido fue un ACA de 0.6680.

1. Introducción

El problema de clasificación se viene estudiando desde los inicios de la visión computacional. A través de los años, han surgido diversos clasificadores, y la manera de estudiar el problema ha cambiado radicalmente. No obstante, el avance de esta área no se debe solo a los nuevos modelos de clasificación, sino también al surgimiento de nuevas y mejores maneras de representar una imagen (los vectores de descripción). A través de este laboratorio, nos fue posible familiarizarnos con una representación muy útil en clasificación: los textones. Se crearon diferentes diccionarios de textones para poder entender como algunos parámetros pueden mejorar la discriminación de los modelos. Adicionalmente, se experimentó con dos modelos de clasificación: 'nearest neighbor' y 'random forest'.

2. Descripción de la base de datos

Para este laboratorio se utilizó la base de datos del grupo Ponce del departamento de Visión por Computador y Robótica de instituto Beckman de la Universidad de Illinois. Esta base de datos consta de 25 clases de texturas con 40 imágenes por clase. Todas las imágenes están en

escala de gris en formato JPG de tamaño 640x480 pixeles [1]. Para este laboratorio se tomaron 756 imágenes para entrenamiento (378 para entrenamiento y validación) y 250 imágenes para test.

3. Descripción de los Métodos y Filtros

Se utilizaron histogramas de textones para representar imágenes de 25 texturas diferentes. Para crear las representaciones (los histogramas), es necesario (1) crear un banco de filtros, (2) crear un diccionario de textones, y después, para cada imagen, (3) filtrar la imagen, (4) asignarle a cada pixel el textón más cercano, y (5) contar cuantos pixeles pertenecen a cada textón (crear histograma).

El procedimiento para crear un diccionario de textones consiste en concatenar un número de imágenes, después filtrar la imagen conjunta con todo el banco de filtro, y finalmente utilizar k-means para encontrar k textones (Los datos de entrada son las respuesta a los filtros de cada pixel). En otras palabras, para crear un diccionario de textones es necesario definir tres parámetros: el banco de filtros, el número de 'palabras', y el número de imágenes a partir de las cuales se calcula el diccionario. El banco de filtros fue creado con la función 'fbCreate', la cual crea 32 fitros. Por otro lado, se trató de optimizar el diccionario de textones experimentando con el número de textones (k), y el número de imágenes utilizadas (n). Se consideraron $k = \{5, 20, 40, 60, 80\}$, y $n = \{5, 10, 20, 30\}$. Los parámetros escogidos, y los resultados de los experimentos, se encuentran en la sección 'Experimentos'.

En la figura 1 aparecen los 32 filtros utilizados. Estos consisten en filtros de dimensión 13x13 y 19x19 con palos (sin transformada de Hilbert) y bordes (con transformada) en 8 direcciones. Por otro lado, las imágenes de texturas clasificadas tenían una gran variedad: algunas tenían texturas grandes, y otras, mucho más finas. Un filtro muy pequeño puede obtener respuesta en ambos tipos de imágenes, mientras que un filtro más grande deja de responder a texturas muy finas. Puesto que un filtro discriminatorio es aquel que obtiene mayor varianza de respuesta en imágenes de diferente textura, se cree que los filtros de 19x19 fueron mejores



Figure 1. Banco de filtro: 32 filtros: 13x13 y 19x19, son y con transformada de Hilbert, 8 orientaciones

que los 13x13. Siguiendo la misma lógica, también se cree que los filtros sin transformada de Hilbert fueron mejores, puesto que éstos solo responden cuando hay líneas negras, mientras que los otros responden a los cambios de intensidad (responden tanto a líneas negras gruesas como a transiciones). Debido a que todas las texturas tienen fotos en múltiples ángulos, no se cree que no haya orientaciones mejores que otras.

4. Descripción de los Clasificadores y Métricas

4.1. Vecino más Cercano

Este método de clasificación utiliza una de las nociones más importantes en la estadística: elementos de la misma clase son más parecidos, es decir están cerca. El método de vecinos más cercanos es el más simple de los clasificadores supervisados. Para este método es necesario considerar entonces las distancias posibles entre los datos. En este laboratorio, dado que se utilizan histogramas de texturas para la clasificación, las distancias utilizadas son la distancia de intersección (ecuación 1) y la distancia chiquadrado (ecuación 2).

$$d(v, w) = \sum_{i=1}^N \min(v_i, w_i) \quad (1)$$

$$d(v, w) = \sum_{i=1}^N \frac{(v_i - w_i)^2}{2(v_i + w_i)} \quad (2)$$

Este método de clasificación calcula las distancias entre el dato a clasificar y todos los ejemplos de entrenamiento. Luego de esto, selecciona los k vecinos más cercanos al dato y asigna la clase más recurrente en dichos vecinos. Vecinos más cercanos es un método muy bueno cuando la muestra de entrenamiento es bastante grande, sin embargo no es tan bueno cuando la muestra es pequeña y es costoso en tiempo para el cálculo de las distancias.

Para este laboratorio consideramos como parámetros a variar la distancia entre los datos (intersección y chiquadrado) y el número de k vecinos a considerar (1 y 3).

4.2. Bosques Aleatorios

Bosques aleatorios es un método que se basa en construir varios árboles de decisión binarios, los cuales se construyen a partir de una muestra de entrenamiento. Su objetivo es construir una separación de la muestra mediante separaciones de los datos a partir de umbrales de las coordenadas de los datos. Para determinar cuál separación es la más apropiada en cada nodo del árbol se define una función de impureza del árbol y se calcula el cambio de esta al agregar una separación dada, con esto se elige aquella separación que más disminuye la función de impureza. La función de impureza es tal que cuanto más diferenciados estén los grupos en las hojas del árbol menor es su valor. De esta manera se construye el árbol hasta que cada hoja del árbol contiene un número de datos mínimos de la muestra. Finalmente se calculan las probabilidades de cada hoja para pertenecer a cada categoría, esto se realiza observando la proporción de cada clase en la hoja de los datos de entrenamiento. De esta manera se construye un solo árbol, para construir entonces un bosque es necesario construir varios árboles diferentes. Para lograr diferencia entre los árboles es necesario limitar el número de datos visible para cada árbol. Para este laboratorio se define entonces el número de coordenadas de cada dato que son utilizadas para construir las separaciones en cada árbol y para cada separación se toma aleatoriamente este número de coordenadas. Finalmente la clase obtenida para un dato de test es la clase que corresponde a la mayor probabilidad ponderada de todos los árboles del bosque.

Para este laboratorio consideramos como parámetro a variar el número de datos mínimos de la muestra que debe haber en cada hoja de los árboles (1, 2, 3, 5 y 10) y el número de árboles del bosque (40, 50, 100, 150, 200, 250 y 300).

4.3. Métrica de Evaluación

Se realizó un modelo de clasificación de imágenes en 25 categorías diferentes. Esto implica que la métrica para evaluar los resultados de la clasificación son la matriz de confusión (Fig. 3) obtenida y el parámetro ACA (average classification accuracy) de exactitud de clasificación promedio, que corresponde al promedio de la diagonal de la matriz de confusión normalizada.

5. Experimentos

5.1. Vecino más Cercano

Los resultados de los experimentos se encuentran en la tabla 1. En esta tabla muestra los ACA de cada experimento. Cada columna representa un diccionario de textones diferente, y cada fila corresponde a una métrica utilizada en el modelo de vecinos más cercanos.

Primero se fijó el número de imágenes a 20 ($n=20$), y se

Table 1. ACA de experimentos vecinos mas cercanos

	n20_k5	n20_k20	n20_k40	n20_k60	n20_k80
k1_Ch2	0.3333	0.5387	0.5987	0.6213	0.6240
kn1_1k	0.3200	0.5573	0.5853	0.6240	0.5947
k3_Ch2	0.3227	0.4880	0.5467	0.5413	0.5333
k3_Ch	0.2960	0.5013	0.5280	0.5227	0.5200

	n5_k60	n10_k60	n20_k60	n30_k60
k1_Ch2	0.6453	0.6293	0.6213	0.6187
kn1_1k	0.6053	0.5973	0.6240	0.5867
k3_Ch2	0.5520	0.5333	0.5413	0.5387
k3_Ch	0.5413	0.5360	0.5227	0.5280

varío el número de 'palabras' ($k=5,20,40,60,80$). Se observó que a medida que aumenta k , aumentó el ACA. Los ACA máximos se obtuvieron con $k = 60$. Al intentar con 80 textones, la precisión disminuyó en la mayoría de las métricas. Esto último se debe a que, si hay demasiados textones, empieza a haber sobreajuste de los datos. Otra observación interesante es que los resultados con solo un vecino fueron mejores que cuando se consideraron tres vecinos, cuando normalmente se espera lo contrario. Esto puede indicar que las clases se confunden fácilmente.

Finalmente, se fijó $k = 60$, y se varió el número de imágenes utilizadas ($n = 5, 10, 20, 30$). Antes de realizar los experimentos, se esperaba que entre más imágenes, mejor sería el resultado, y nos sorprendimos al descubrir que lo contrario es cierto. Se cree que esto se puede deberse al k -means, que puede converger con mayor facilidad si hay pocos datos. Por lo tanto, es posible que un menor número de imagen permita obtener textones más coherentes y, de ahí, mejores resultados.

El mejor método, y el utilizado para 'test' fue utilizando diccionarios con $n=5$ y $k=60$, y utilizando la distancia chi-cuadrado con un solo vecino.

5.2. Bosques Aleatorios

Los experimentos realizados para el clasificador de Bosques Aleatorios se observan en la tabla 2. Para cada tipo de histograma de textones se varió tanto el número de árboles del bosque, para valores de 40, 50, 100, 150, 200, 250, 300 árboles; y se varió también el número de muestras mínimas de entrenamiento en cada hoja para la construcción de cada árbol, para valores de 1, 2, 3, 5, 10. Se puede observar en la tabla 2 una tendencia en cada tipo de histogramas a un aumento del ACA a medida que el número de árboles crece y un decrecimiento de ACA a medida que el número mínimo de muestras aumenta.

Dados los resultados de los otros tipos de histograma se puede observar al comparar Histes _n5 _k40 y Histes _n20 _k40 que a pesar de tener un número de imágenes distintas

Table 2. ACA de Experimentos de Bosques Aleatorios

Histal_n5_k40		Número de árboles						
		40	50	100	150	200	250	300
Min muestras por hoja	1	0.669	0.675	0.701	0.707	0.699	0.701	0.715
	2	0.677	0.667	0.661	0.709	0.704	0.712	0.709
	3	0.659	0.664	0.693	0.696	0.704	0.693	0.683
	5	0.688	0.648	0.677	0.677	0.677	0.675	0.675
	10	0.627	0.608	0.632	0.648	0.640	0.635	0.648

Histes_n5_k40		Número de árboles						
		40	50	100	150	200	250	300
Min muestras por hoja	1	0.653	0.667	0.683	0.672	0.693	0.699	0.696
	2	0.664	0.664	0.669	0.675	0.677	0.699	0.688
	3	0.627	0.664	0.640	0.685	0.664	0.683	0.693
	5	0.627	0.632	0.648	0.667	0.677	0.664	0.688
	10	0.605	0.608	0.624	0.613	0.627	0.637	0.624

Histes_n20_k5		Número de árboles						
		40	50	100	150	200	250	300
Min muestras por hoja	1	0.387	0.385	0.366	0.360	0.368	0.377	0.376
	2	0.371	0.358	0.374	0.363	0.366	0.358	0.374
	3	0.360	0.355	0.374	0.363	0.374	0.368	0.366
	5	0.347	0.355	0.355	0.368	0.349	0.363	0.363
	10	0.355	0.336	0.360	0.347	0.347	0.344	0.344

Hists_n20_k20		Número de árboles						
		40	50	100	150	200	250	300
Min muestras por hoja	1	0.596	0.587	0.610	0.616	0.608	0.604	0.605
	2	0.603	0.573	0.595	0.605	0.608	0.608	0.616
	3	0.565	0.616	0.597	0.589	0.605	0.592	0.600
	5	0.557	0.555	0.584	0.573	0.587	0.584	0.589
	10	0.533	0.547	0.557	0.563	0.555	0.565	0.549

Histes_n20_k40		Número de árboles						
		40	50	100	150	200	250	300
Min muestras por hoja	1	0.675	0.672	0.685	0.680	0.688	0.693	0.691
	2	0.664	0.667	0.688	0.691	0.691	0.688	0.693
	3	0.632	0.667	0.661	0.693	0.680	0.667	0.685
	5	0.635	0.635	0.659	0.653	0.659	0.672	0.659
	10	0.568	0.613	0.632	0.619	0.624	0.635	0.640

Histes_n20_k60		Número de árboles						
		40	50	100	150	200	250	300
Min muestras por hoja	1	0.649	0.665	0.666	0.681	0.682	0.682	0.673
	2	0.666	0.636	0.657	0.666	0.671	0.663	0.665
	3	0.657	0.647	0.647	0.668	0.663	0.658	0.652
	5	0.601	0.628	0.641	0.649	0.625	0.641	0.615
	10	0.579	0.609	0.596	0.572	0.597	0.605	0.587

con las que se construye el banco de filtros los valores en la clasificación no varían mucho. Así mismo se puede observar con los resultados de Histes _n20 _k5, Histes _n20 _k20 y Histes _n20 _k40 que a medida que el número de bins aumenta la clasificación obtenida es mejor. Sin embargo se observa, tomando Histes _n20 _k60 en cuenta, que existe un punto óptimo local para el número de bins entre un k de 40 y una de 60, esto porque los valores con k de 60 son menores que los de Histes _n20 _k40.

Por otro lado el mejor método, obtenido en el conjunto de validación, corresponde al histograma de textones que no

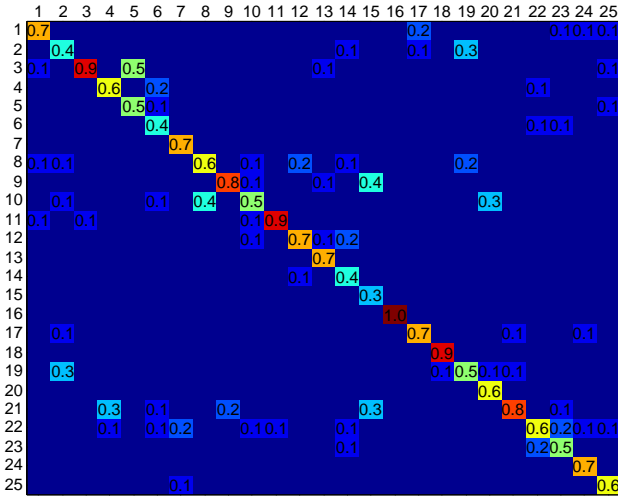


Figure 2. Banco de filtro: 32 filtros: 13x13 y 19x19, son y con transformada de Hilbert, 8 orientaciones

fue construido aleatoriamente, sino que consideraron las primeras 5 imágenes para realizar el banco de textones y se consideraron 40 bins para los histogramas. Este tipo de histograma obtuvo para 300 árboles en el bosque con 1 muestra mínima por hoja un ACA de 71.5% y para 250 árboles con 2 muestras mínimas por hoja un ACA de 71.2%. El método elegido para realizar la clasificación fue utilizando el banco de textones de 5 imágenes con 40 bins, 300 árboles en el bosque con 1 muestra mínima por hoja .

6. Resultados

Al entrenar el modelo de vecinos más cercano con todas las imágenes de train, y evaluarlo en test, se obtuvo la matriz de confusión de la figura 2. Esto nos resulta en un ACA de 0.6400. Adicionalmente, el entrenamiento y evaluación (sin contar el cálculo de los vectores de representación) fue de 0.1901 segundos

Los resultados para el método de clasificación de bosques aleatorios en los datos de test se muestra en la matriz de confusión 3 con un ACA de 0.6680. En la figura 4 se muestra la matriz de clasificación de las imágenes de test cuya intensidad corresponde al valor de confianza obtenido para la categoría elegida por el clasificador. Este código tardó 22.4657 en correr.

7. Discusión

Comprando los puntajes ACA de ambos métodos, observamos 0.64 para vecinos más cercanos y 0.668 para árboles. Por lo tanto, los árboles aleatorios resolvieron mejor este problema. No obstante, al comparar los tiempos

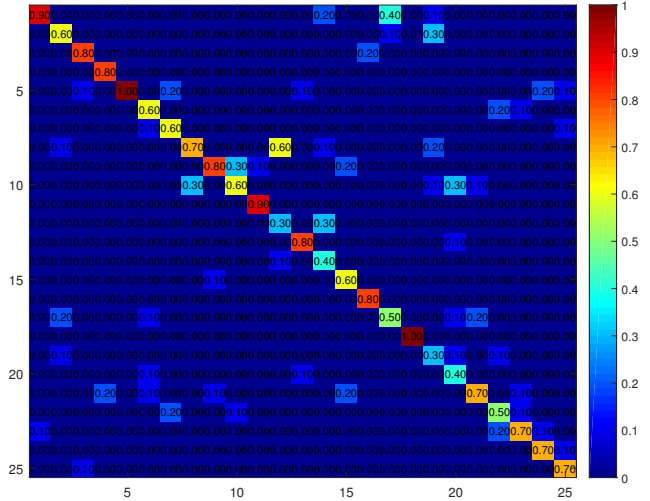


Figure 3. Resultados Bosques Aleatorios. Matriz de confusión

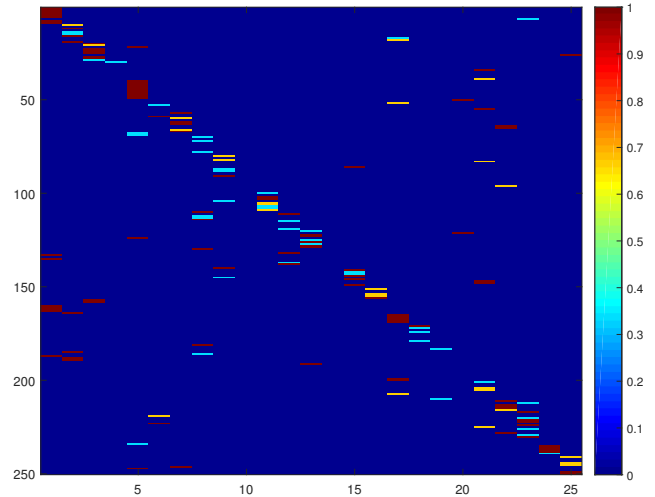


Figure 4. Resultados por imagen de test con el método de bosques aleatorios. Las intensidades corresponden a la confianza con la que el clasificador decide la clase a la que pertenecen. Las imágenes están organizadas en escala ascendente, en donde el ideal de clasificación corresponde únicamente a la diagonal de la matriz.

de ejecución, notamos que el clasificador de vecinos más cercanos es mucho más rápido (0.1901 contra 22.4657). Por una diferencia en precisión de 0.028, un tiempo más de 100 veces mayor puede que no sea recomendable.

Al observar la matriz de confusión de vecinos más cercanos 2 es posible identificar aquellas clases mal clasificadas, tenemos entonces la clase 3 y 5, 10 y 8, 9 y 15 y 21, 19 y 2. Por otro lado se pueden determinar las clases casi perfectamente clasificadas como lo son la clase 3, 11, 16 y 18.

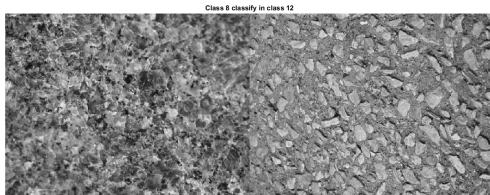


Figure 5. Comparación entre clases mal clasificadas. Lado izquierdo clase 8 lado derecho clase 12.

Según los resultados obtenidos al observar la matriz de confusión del método de árboles aleatorios 3 es posible identificar cuáles categorías son aquellas más mal clasificadas. Uno de las clases que obtuvo mayor error en clasificación fue la categoría 8 que fue clasificada en la 12. Esto se debe a que ambas texturas son similares (figura 5). Otras categorías con bastante confusión para FR fueron la 17 y 1, 14 y 12, la 19 y 2, la 10 y 20. Así mismo podemos determinar las clases casi perfectamente clasificadas como lo son la 1, 5, 11, 18.

Se cree que las clases mal clasificadas entre métodos varían debido a que la distancia usada en cada método de clasificación utiliza una distancia diferente lo que hace que tengan cercanías definidas diferentes.

8. Limitaciones

Una de las principales limitaciones para el método de árboles aleatorios fue elegir la combinación de parámetros a variar para optimizar el método. Esto debido a que la cantidad de parámetros que tiene este tipo de clasificador es muy alto. Dado el tiempo limitado para realizar más experimentos e incluir más parámetros variables uno de las mejoras sería afectivamente realizar pruebas variando otros parámetros como el número de coordenadas visibles en cada separación de cada árbol, el número min de muestras de la muestra para realizar una nueva separación, considerar posibles podas de árboles, etc.

Por otro lado, vecinos más cercanos tiene pocos parámetros, por lo que es fácil realizar experimentos. No obstante, aun con parámetros óptimos, no se pudo superar a los árboles, puesto que vecinos-más-cercanos es muy sensible a clases parecidas.

La base de datos, por su parte, también tiene un par de limitaciones. Por un lado, el número de imágenes no es muy grande. Por otro lado, cada categoría solo tiene

fotos de un mismo objeto. Efectivamente hay cambios de escala y ángulo, pero el problema sería más interesante si, por ejemplo, la clase de textura ladrillo tuviera fotos de diferentes paredes de ladrillos.

9. Mejoras

A lo largo de éste laboratorio, se experimentó con los parámetros de los clasificadores, y el número de textones e imágenes en el diccionario de textones. No obstante, falta variar el banco de filtro, para ajustarlo a la base de datos utilizada. En especial, hay varias texturas que no presentan rayas, sino manchas, por lo cual la adición de un filtro circular podría ser de gran ayuda. También comparar resultados con otros bancos de filtros, y considerar escalas diferentes de la imagen.

Otra posible mejora es aumentar la base de entrenamiento. Para esto, se pueden voltear las imágenes de derecha-izquierda y arriba-abajo, de manera que se tendría cuatro veces las imágenes que se tienen ahora. Esto sería de gran ayuda, en especial, para el clasificar de vecino más cercano, que es ideal cuando el número de ejemplos es infinito.

Finalmente, es posible considerar un preprocesamiento de las imágenes, como un aumento de contraste, que podría resultar en una mayor distinción entre las clases de textura, y por lo tanto, facilitaría su clasificación correcta.

References

- [1] C. S. Svetlana Lazebnik and J. Ponce. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278, 2005.