PRAC 5

SUPPORT VECTOR MACHINES

CLP: Clasificación de Patrones.
ETSETB
UPC

Octubre 2016

1 Objetivos

Aplicación de las técnicas de clasificación con Support Vector Machine (lineal y no-lineal). Cálculo de varias figuras de calidad del clasificador obtenido. Evaluación de la validez de decisiones en clasificadores.

2 Base de datos

Para trabajar con SVM adoptaremos un problema de clasificación con dos clases. La base de datos SPAM se halla formada por 4601 vectores obtenidos de 4601 e-mails. A partir de la frecuencia con que aparece cada palabra se debe predecir si el e-mail es SPAM o no lo es. Cada vector es de 57 coordenadas.

Esta es la información que los autores de la base de datos facilitan sobre la misma:

1. Title: SPAM E-mail Database

2. Sources:
(a) Creators: Mark Hopkins, Erik Reeber, George Forman, Jaap Suermondt Hewlett-Packard Labs, 1501 Page Mill Rd., Palo Alto, CA 94304
(b) Donor: George Forman (gforman at nospam hpl.hp.com) 650-857-7835
(c) Generated: June-July 1999

3. Past Usage:
(a) Hewlett-Packard Internal-only Technical Report. External forthcoming.
(b) Determine whether a given email is spam or not.
(c) ~7% misclassification error.
False positives (marking good mail as spam) are very undesirable.
If we insist on zero false positives in the training/testing set, 20-25% of the spam passed through the filter.

4. Relevant Information:
" The ""spam"" concept is diverse: advertisements for products/web" sites, make money fast schemes, chain letters, pornography...Our collection of spam e-mails came from our postmaster and individuals who had filed spam. Our collection of non-spam e-mails came from filed work and personal e-mails, and hence the word 'george' and the area code '650' are indicators of non-spam. These are useful when constructing a personalized spam filter. One would either have to blind such non-spam indicators or get a very wide collection of non-spam to generate a general purpose spam filter.

For background on spam:
Cranor, Lorrie F., LaMacchia, Brian A. Spam!
Communications of the ACM, 41(8):74-83, 1998.

5. Number of Instances: 4601 (1813 Spam = 39.4%)

6. Number of Attributes: 58 (57 continuous, 1 nominal class label)

7. Attribute Information: The last column of 'spambase.data' denotes whether the e-mail was considered spam (1) or not (0), i.e. unsolicited commercial e-mail. Most of the attributes indicate whether a particular word or character was frequently occuring in the e-mail. The run-length attributes (55-57) measure the length of sequences of consecutive capital letters. For the statistical measures of each attribute, see the end of this file. Here are the definitions of the attributes:

48 continuous real [0,100] attributes of type word_freq_WORD = percentage of words in the e-mail that match WORD, i.e. 100 * (number of times the WORD appears in the e-mail) / "total number of words in e-mail. A ""word"" in this case is any " string of alphanumeric characters bounded by non-alphanumeric characters or end-of-string.

6 continuous real [0,100] attributes of type char_freq_CHAR = percentage of characters in the e-mail that

match CHAR, i.e. 100 * (number of CHAR occurences) / total characters in e-mail

1 continuous real [1,...] attribute of type capital_run_length_average == average length of uninterrupted sequences of capital letters

1 continuous integer [1,...] attribute of type capital_run_length_longest =length of longest uninterrupted sequence of capital letters

1 continuous integer [1,...] attribute of type capital_run_length_total = = sum of length of uninterrupted sequences of capital letters = total number of capital letters in the e-mail

1 nominal {0,1} class attribute of type spam = denotes whether the e-mail was considered spam (1) or not (0), i.e. unsolicited commercial e-mail.

8. Missing Attribute Values: None

9. Class Distribution:
Spam 1813 (39.4%)
Non-Spam 2788 (60.6%)

This file: 'spambase.DOCUMENTATION' at the UCI Machine Learning Repository
https://archive.ics.uci.edu/ml/datasets/Spambase

NEXT TABLE shows the content of feature vector (dimension = 57)

Number	Feature	Number	Feature
1	word_freq_make: continuous.	30	word_freq_labs: continuous.
2	word_freq_address: continuous.	31	word_freq_telnet: continuous.
3	word_freq_all: continuous.	32	word_freq_857: continuous.
4	word_freq_3d: continuous.	33	word_freq_data: continuous.
5	word_freq_our: continuous.	34	word_freq_415: continuous.
6	word_freq_over: continuous.	35	word_freq_85: continuous.
7	word_freq_remove: continuous.	36	word_freq_technology: continuous.
8	word_freq_internet: continuous.	37	word_freq_1999: continuous.
9	word_freq_order: continuous.	38	word_freq_parts: continuous.
10	word_freq_mail: continuous.	39	word_freq_pm: continuous.
11	word_freq_receive: continuous.	40	word_freq_direct: continuous.
12	word_freq_will: continuous.	41	word_freq_cs: continuous.
13	word_freq_people: continuous.	42	word_freq_meeting: continuous.
14	word_freq_report: continuous.	43	word_freq_original: continuous.
15	word_freq_addresses: continuous.	44	word_freq_project: continuous.
16	word_freq_free: continuous.	45	word_freq_re: continuous.
17	word_freq_business: continuous.	46	word_freq_edu: continuous.
18	word_freq_email: continuous.	47	word_freq_table: continuous.
19	word_freq_you: continuous.	48	word_freq_conference: continuous.
20	word_freq_credit: continuous.	49	char_freq_;: continuous.
21	word_freq_your: continuous.	50	char_freq_(: continuous.
22	word_freq_font: continuous.	51	char_freq_[: continuous.
23	word_freq_000: continuous.	52	char_freq_!: continuous.
24	word_freq_money: continuous.	53	char_freq_$: continuous.
25	word_freq_hp: continuous.	54	char_freq_#: continuous.
26	word_freq_hpl: continuous.	55	capital_run_length_average: continuous.
27	word_freq_george: continuous.	56	capital_run_length_longest: continuous.
28	word_freq_650: continuous.	57	capital_run_length_total: continuous.
29	word_freq_lab: continuous.		

3 A realizar en el laboratorio

La base de datos utilizada en esta práctica se divide en tres subconjuntos, como se puede observar en el software facilitado. Los tres subconjuntos son:

- Base de datos de train o entreno: Conjunto de vectores y etiquetas que se utiliza para entrenar y obtener cada clasificador con unos determinados parámetros. Se debe entrenar un clasificador para cada valor de un conjunto de posibles valores de un parámetro dado. Así para un parámetro se propone evaluar desde V1, V2, .. hasta VN y se obtienen N clasificadores distintos. Si existe más de un parámetro, se debe entrenar un clasificador para cada combinación posible de valores de parámetros. Por ejemplo, con dos parámetros de N posibles valores cada uno, se deberían entrenar N^2 clasificadores.
- Base de datos de validation o validación: Conjunto de vectores y etiquetas que se utiliza para medir el error de clasificación a partir de cada uno de los clasificadores entrenados con la base de datos de entreno. De este modo, el clasificador que proporcione un error menor de validación, es el que determina el valor del parámetro Vi óptimo.
- Base de datos de test: Conjunto de vectores y etiquetas que se utiliza para medir el error de clasificación únicamente con el clasificador óptimo seleccionado con la base de datos de validación. Los errores medidos sobre esta base de datos son los que determinan la calidad del clasificador diseñado.

3.1 Obtención del clasificador SVM

En este apartado se va a estudiar la influencia del parámetro de regularización P en un clasificador SVM. El parámetro de regularización P es un valor real y positivo que controla la penalización adjudicada a los vectores de entrenamiento clasificados erróneamente. Un valor alto de P conduce a un clasificador sobreentrenado además de ralentizar excesivamente el tiempo de obtención del clasificador. Por otro lado, un valor excesivamente bajo del parámetro P, conduce a un clasificador infraentrenado y con prestaciones de baja calidad. Puede consultar la presentación explicada en clase de teoría para más información sobre dicho parámetro.

1. Edite el fichero *prac5_SPAM.m* y observe todas las etapas del script identificando las siguientes partes:

- Simplificación de la base de datos a base de eliminar las características 55, 56 y 57 y de cuantificar el resto de características de forma binaria. Es decir, si la característica m-ésima de un vector es igual a 1, significa que el correspondiente e-mail contiene al menos una vez la palabra específica de la fila m de la tabla del apartado 2.1 mientras que en caso contrario la correspondiente característica es igual a 0.
- División de la base de datos en train (60%), validation (20%) y test (20%) previo aislamiento de un vector de la base de datos junto con su correspondiente etiqueta al que se denomina V_analisis.
- Aplicación de clasificador svm lineal, con parámetro de regularización P=0.1.
- Aplicación de clasificador no lineal con Kernel Gaussiano de parámetro de escalado *h*=1 y parámetro de regularización P=0.1.

2. Ejecute el fichero *prac5_SPAM* y anote los valores obtenidos para los errores de clasificación de train y de test, con clasificador lineal y con clasificador de kernel gaussiano y comente los resultados. No olvide anotar el valor de los parámetros P y de *h* que se han utilizado en esta parte.

3. Para obtener los valores del parámetro de regularización P óptimo y de parámetro de escalado *h* óptimo en el caso de Kernel Gaussiano, diseñe con la base de datos de train y mida el error obtenido tanto en train como en validation al variar ambos parámetros para los siguientes valores P = 0.01:0.1:5 y *h* = [1 2.5 25 100]. Asegúrese al programar el doble bucle, que conserva el clasificador correspondiente al menor error de validation y que lo podrá utilizar en el resto de la práctica sin tener que volver a generarlo. Obtenga una gráfica bidimensional con los dos errores de clasificación (train y validation) en función del parámetro P y del parámetro *h*. Copie el código generado y la gráfica obtenida en la memoria y decida los valores óptimos de P y de *h* a la vista de los resultados.

4. Con el clasificador óptimo hallado en el apartado anterior mida el error de clasificación sobre la base de test y compare de nuevo con el que había obtenido con el clasificador gaussiano no optimizado (punto 2 de este apartado).

3.2 Análisis de la bondad del clasificador

En el cálculo de probabilidades de error con dos clases han de tenerse en cuenta los siguientes indicadores:

FP (False Positives): # de Falsos positivos o en el caso que nos ocupa elementos clasificados como SPAM cuando realmente son MAIL.
TP (True Positives): # de Verdaderos positivos o en el caso que nos ocupa elementos clasificados como SPAM cuando realmente son SPAM.
FN (False Negatives): # de Falsos negativos o en el caso que nos ocupa elementos clasificados como MAIL cuando realmente son SPAM.
TN (True Negatives): # de Verdaderos negativos o en el caso que nos ocupa elementos clasificados como MAIL cuando realmente son MAIL.

El **Error de clasificación** se puede expresar como $E_c = \frac{FP + FN}{FP + TP + FN + TN}$

El indicador de calidad complementario al error es la denominada **Accuracy** y se define como: $A = 1 - E_c = \frac{TP + TN}{FP + TP + FN + TN}$

En ocasiones estos indicadores de calidad no indican correctamente si el clasificador funciona de forma adecuada, especialmente cuando se tienen muchos más elementos de una clase respecto a la otra. En su lugar se pueden utilizar las medidas de **Precisión**, **Sensiblidad** (o **Recall**), **Especificidad** y **F_score** definidas a continuación. Observe que en todos los casos interesa que sean de valor próximo a 1 para un clasificador que funcione adecuadamente.

$$\mathrm{P} = \frac{\text{Clasificado SPAM correctamente}}{\text{Clasificado SPAM}} = \frac{TP}{TP+FP}$$

$$S = R = \frac{\text{Clasificado SPAM correctamente}}{\text{\# total de SPAM}} = \frac{TP}{TP+FN}$$

$$Es = \frac{\text{Clasificado MAIL correctamente}}{\text{\# total de MAIL}} = \frac{TN}{TN+FP}$$

$$F\_score = 2\frac{P.R}{P+R}$$

A partir de este punto se utilizará el clasificador de kernel gaussiano óptimo obtenido en el punto anterior para predecir la clase, SPAM o MAIL sobre la base de datos de test, separándola cuando convenga en conjunto de vectores de SPAM y conjunto de vectores de MAIL, según lo que se haya de calcular.

1. Halle los 6 indicadores ($E_c, A, P, S, E_s, F\_score$) sobre la base de datos de test y con el clasificador óptimo obtenido en el apartado 3.1.
2. Justifique para una BD compuesta por 400 vectores de tipo SPAM y 4600 vectores de tipo MAIL, porqué los cocientes P, S, E_s, F_score resultan más adecuados que E_c, A para medir la bondad del clasificador.

3.3 Análisis de la validez de decisión

1. Aplique el clasificador no lineal óptimo obtenido en el punto anterior a la base de datos de test y halle las probabilidades a priori:

$$\Pr\{\text{clase=SPAM}\} = \frac{\#\text{elementos SPAM de la BD de test}}{\#\text{elementos de la BD de test}}$$

$$\Pr\{\text{clase=MAIL}\} = \frac{\#\text{elementos MAIL de la BD de test}}{\#\text{elementos de la BD de test}}$$

2. Clasifique el vector aislado (V_analisis). Comente si se clasifica correctamente o incorrectamente. ¿Contenía el e-mail correspondiente la palabra "make"? ¿Y la palabra "address"?

3. Determine la validez de la decisión. Para ello, si el vector aislado se clasifica como SPAM halle la probabilidad de que sea SPAM realmente : $\Pr\{\text{V\_analisis=SPAM} \mid \text{clasificador=SPAM}\}$ o bien si el vector aislado se clasifica como MAIL, halle la probabilidad de que sea MAIL realmente: $\Pr\{\text{V\_analisis=MAIL} \mid \text{clasificador=MAIL}\}$. Utilice la base de datos de test y los resultados proporcionados por el clasificador cómo crea conveniente, y explique de forma detallada el método utilizado. Incluya el código generado para obtener la probabilidad solicitada en este punto.

NOTA IMPORTANTE

Las explicaciones deben apoyarse en los conceptos teóricos vistos en clase. El documento de entrega de práctica debe incluir todos los resultados gráficos y numéricos y explicaciones que se consideren convenientes, es decir, aquellos que aporten información adicional y relevante.