

Práctica 5

Clasificación Bayesiana

Javier Herrer Torres (NIP: 776609)

Aprendizaje automático
Grado en Ingeniería Informática



**Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza**

Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza
Curso 2020/2021

1. Objetivo

El objetivo es resolver mediante clasificación Bayesiana un problema real de clasificación multi-clase: el reconocimiento de dígitos manuscritos. Igual que en la práctica anterior, utilizaremos una versión reducida del conjunto de datos MNIST, y usaremos para la clasificación los niveles de intensidad de los 400 píxeles de cada dígito. Utilizaremos modelos Gaussianos para los atributos de cada clase.

2. Estudio previo

En la sección 3 se ha escrito el algoritmo de entrenamiento y clasificación multi-clase utilizando clasificación Bayesiana con atributos Gaussianos. Para su implementación, se han empleado las transparencias del tema de Modelos Generativos, concretamente:

$$N_j = \sum_i [y^{(i)} = j]$$

$$\mu_j = \frac{1}{N_j} \sum_{y^{(i)}=j} x^{(i)}$$

$$\Sigma_j = \frac{1}{N_j - 1} \sum_{y^{(i)}=j} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T$$

Por otro lado, se ha comprobado que existen suficientes datos de entrenamiento. De lo contrario se podría sufrir fácilmente de sobreajuste porque el número de parámetros que se aprendan sea muy grande comparado con el número de muestras.

El número de muestras para cada clase j es $N_j = 400$, y el número de parámetros que se aprenden es $D = \text{size}(\mu_j) = 401$. Para evitar el sobreajuste se puede emplear Bayes ingenuo o regularización en la estimación de las covarianzas.

3. Entrenamiento y clasificación con modelos Gaussianos regularizados

Aplicando regularización, las matrices de covarianza ya no son las calculadas con Matlab mediante `cov` sino otras con términos en la diagonal. Garantizando que Σ es invertible y calculando bien la distancia de Mahalanobis y la densidad de la Gaussiana. Es decir, **es imprescindible regularizar**.

$$\Sigma'_j = \Sigma_j + \lambda I_D$$

```

function modelo = entrenarGaussianas( Xtr, ytr, nc, NaiveBayes, landa )
% Entrena una Gaussiana para cada clase y devuelve:
% modelo{i}.N      : Numero de muestras de la clase i
% modelo{i}.mu     : Media de la clase i
% modelo{i}.Sigma  : Covarianza de la clase i
% Si NaiveBayes = 1, las matrices de Covarianza serán diagonales
% Se regularizarán las covarianzas mediante: Sigma = Sigma + landa*eye(D)

% Transparencia 17 Modelos Generativos
for j = 1:nc
    muestras = ytr == j;
    N_j = sum(muestras);
    modelo{j}.N = N_j;

    x_j = Xtr(muestras, :);
    media = mean(x_j);
    modelo{j}.mu = media;

    covarianza = cov(x_j);
    modelo{j}.Sigma = covarianza;

    % Transparencia 23 Modelos Generativos
    if NaiveBayes == 1
        % Bayes ingenuo
        modelo{j}.Sigma = diag(diag(covarianza));
    end
    % Regularización en la estimación de las covarianzas
    modelo{j}.Sigma = modelo{j}.Sigma + landa * eye(length(media));
end

function yhat = clasificacionBayesiana( modelo, X)
% Con los modelos entrenados, predice la clase para cada muestra X
pred = [];
for clase = 1:10
    y = gaussLog(modelo{clase}.mu, modelo{clase}.Sigma, X);
    pred = [pred y];
end
[~, yhat] = max(pred, [], 2);

```

4. Bayes ingenuo

Se ha programado el entrenamiento y clasificación multi-clase, usando Bayes ingenuo, basándose en el código de la práctica anterior. Para ello, se ha separado un 20% de los datos para validación ($k = 5$), y se ha encontrado el mejor valor para el parámetro de regularización.

$$best_lambda = 0,0078$$

En la figura 1 se encuentra dibujada la gráfica de tasa de errores en función del parámetro λ . Se puede observar sub-ajuste a la derecha, y sobre-ajuste a la izquierda del punto ideal que se encuentra en:

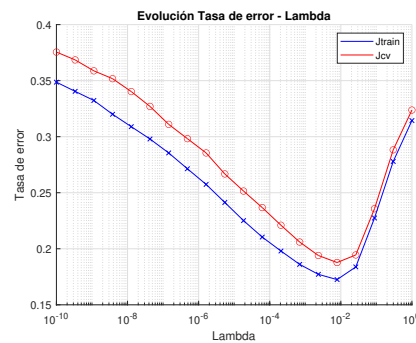


Figura 1: Gráfica de tasa de errores en función del parámetro λ

Posteriormente, se ha entrenado con ese mejor valor de λ y se han obtenido las siguientes tasas de error:

$$E_{train}(\theta) = 0,1742$$

$$E_{test}(\theta) = 0,1950$$

4.1. Matriz de confusión y Precision/Recall

Se ha re-entrenado con todos los datos para el mejor valor de λ , y se han utilizado los datos de test para calcular la matriz de confusión de la figura 5 junto con los valores de precisión y recall para cada dígito, y los globales. En la figura 6 se pueden apreciar dígitos problemáticos:

- El dígito 2 se confunde con el 6 y con el 8.
- El dígito 4 se confunde con el 9.
- El dígito 5 se confunde con el 3 y con el 8.
- El dígito 8 se confunde con el 1 y con el 9.

De los que damos como positivos, el 81,61 % lo son realmente. De los casos positivos, detectamos el 80,50 %.

	Dígito Predicho										Suma	Recall
	1	2	3	4	5	6	7	8	9	0		
Dígito Real	1	98	1	1							100	0,98
	2	5	76			9		8		2	100	0,76
	3	5	6	72	1	2	1	7	5		100	0,72
	4	1	4		71	1	3		3	17	100	0,71
	5	3	1	11	4	59	5	1	7	5	100	0,59
	6	2	1			95		2			100	0,95
	7	4			4		83	1	8		100	0,83
	8	7	2	3	3	3	1	74	7		100	0,74
	9	2		1	4		2	3	88		100	0,88
	0				3	4			4	89	100	0,89
Suma	127	91	88	87	68	118	87	109	130	95	1000	0,8050
Precision	0,77	0,84	0,82	0,82	0,87	0,81	0,95	0,68	0,68	0,94	0,8161	

Figura 2: Matriz de confusión

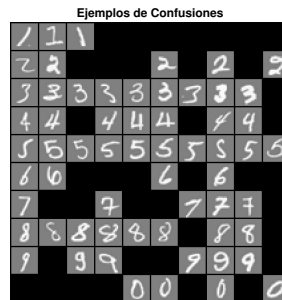


Figura 3: Confusiones

5. Covarianzas completas

Se ha programado el entrenamiento y clasificación multi-clase, usando matrices de covarianzas completas. Para ello, se ha separado un 20 % de los datos para validación ($k = 5$), y se ha encontrado el mejor valor para el parámetro de regularización.

$$best_ \lambda = 0,0264$$

En la figura 4 se encuentra dibujada la gráfica de tasa de errores en función del parámetro λ . Se puede observar sub-ajuste a la derecha, y sobre-ajuste a la izquierda del punto ideal que se encuentra en:

Posteriormente, se ha entrenado con ese mejor valor de λ y se han obtenido las siguientes tasas de error:

$$E_{train}(\theta) = 0,0123$$

$$E_{test}(\theta) = 0,0390$$

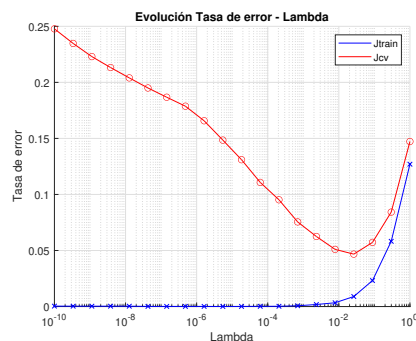


Figura 4: Gráfica de tasa de errores en función del parámetro λ

5.1. Matriz de confusión y Precision/Recall

Se ha re-entrenado con todos los datos para el mejor valor de λ , y se han utilizado los datos de test para calcular la matriz de confusión de la figura 5 junto con los valores de precisión y recall para cada dígito, y los globales. En la figura 6 se pueden apreciar dígitos problemáticos:

- El dígito 3 se confunde con el 2.
- El dígito 5 se confunde con el 8.
- El dígito 8 se confunde con el 3.

De los que damos como positivos, el 96,15 % lo son realmente. De los casos positivos, detectamos el 96,10 %.

	Dígito Predicho										Suma	Recall
	1	2	3	4	5	6	7	8	9	0		
Dígito Real	1	99	1								100	0,99
	2	1	97	1				1			100	0,97
	3		3	94	1			1	1		100	0,94
	4	1	1		96					1	100	0,96
	5			1		94	1		4		100	0,94
	6						99			1	100	0,99
	7	2						96		2	100	0,96
	8	2	1	3		1			92	1	100	0,92
	9	1		2			1	2		94	100	0,94
	0									100	100	1
	Suma	106	103	101	96	96	100	97	100	99	1002	0,9610
	Precision	0,93	0,94	0,93	1,00	0,98	0,99	0,99	0,92	0,95	0,98	0,9615

Figura 5: Matriz de confusión

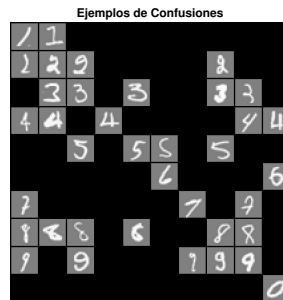


Figura 6: Confusiones

6. Comparación de modelos

Cabe destacar de nuevo la importancia de la regularización con el ejemplo visto en clase. Hay píxeles que «no sirven para nada» y que en todas las muestras de entrenamiento de un dígito, por ejemplo el 3, no hay ningún valor para ese píxel. Pero, un 3 que tuviese una mancha en ese punto no habría ninguna posibilidad de que fuese un 3 si no se aplicara *suavización*.

En la figura 7 se puede apreciar que existe una clara correlación positiva en la diagonal de la matriz de covarianza. Esto se debe a la *vecindad en el trazo*, es decir, si pinto un píxel con el boli es muy probable que los píxeles vecinos lo estén también.

Esta es la razón por la que se obtienen unos peores resultados con Bayes ingenuo. Este modelo se debe emplear en el caso contrario, cuando los atributos sean independientes, es decir, no estén correlados. En la figura 8 se aprecia la pérdida de precisión de Bayes ingenuo.

La diferencia entre el modelo con Covarianzas completas y con Regresión Logística se debe al empleo de un modelo generativo que suele hacer suposiciones adicionales que, si son acertadas, suelen aprender más deprisa que los modelos discriminativos. Un buen ejemplo es el reconocimiento con descriptores.

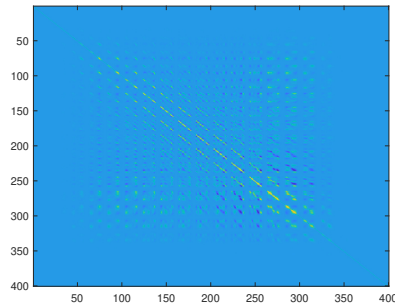


Figura 7: Correlaciones Clase 3

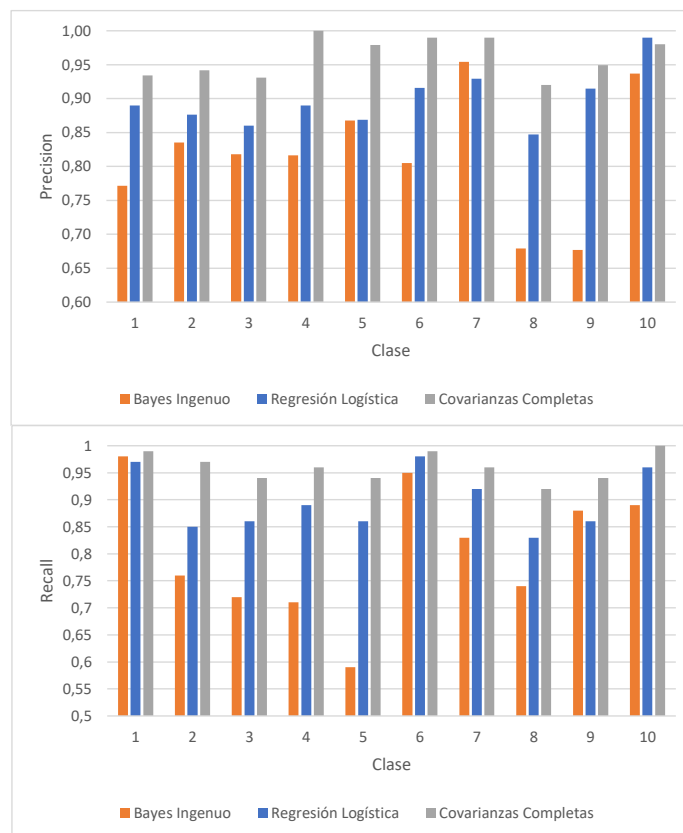


Figura 8: Comparación de modelos