

## **Asignatura: Extracción de características en imágenes**

**Curso 2018-2019**

## **Máster: Ciencia de Datos e Ingeniería de Computadores**

### **Práctica III: Uso de Procesos Gaussianos para clasificación**

El objetivo de esta práctica es que aprendas a utilizar los Procesos Gaussianos (GP) en un problema de clasificación y discutas los resultados que hayas obtenido.

Vamos a utilizar una base de datos de imágenes histológicas de cáncer de próstata. La figura 1 contiene varias imágenes de tejido histológico sano y cancerígeno. Con el uso de microscopio, las imágenes son observables a diferentes resoluciones.

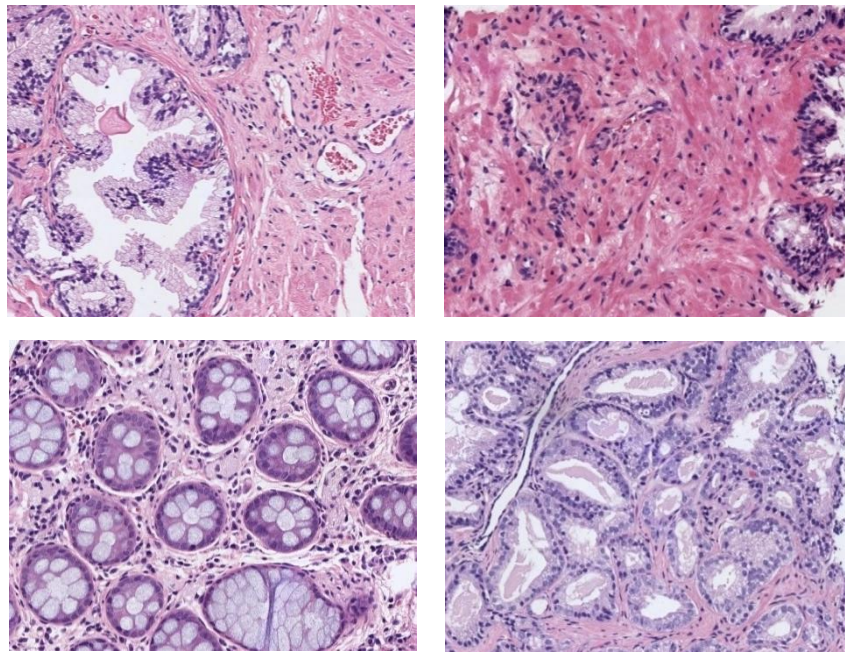


Figura 1. Imágenes de tejido sano (panel superior) y de tejido cancerígeno (panel inferior).

Para abordar el problema de clasificación se han utilizado bloques. Un problema importante es decidir su tamaño. Cada muestra o instancia corresponde a un bloque de tamaño 2048x2048 y ha sido clasificado por un anatomopatólogo como cancerígeno o no cancerígeno.

Para cada bloque 2048x2048 hemos calculado el histograma de los rasgos obtenidos tras calcular características LBP uniformes invariantes por rotaciones con radio 1 y número de vecinos igual a 8 sobre cada uno de los píxeles del parche. Por tanto, cada bloque queda resumido por un histograma sobre 10 valores (0 a 9) que cuenta el número de píxeles en el bloque a los que se le asigna la misma etiqueta (un número entre 0 y 9).

Las siguientes imágenes explican cómo una etiqueta de 0 a 9 fue asignada a cada píxel, conocidos sus vecinos. En la figura 2, los puntos blancos corresponden a píxeles con valores mayores que el central y los negros a píxeles con valores menores o iguales. El número en el centro indica el correspondiente código LBP.

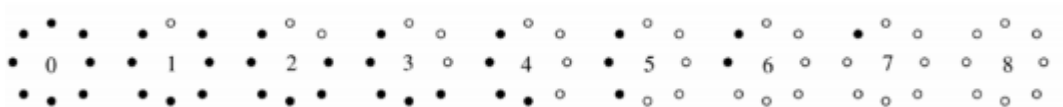


Figura 2.

Observa en los gráficos más a la izquierda y derecha de la figura 2 que a los dos patrones con 0 transiciones 0-1 le fue asignado o bien el 0 o bien el 8. En la misma figura, a los distintos patrones (y sus rotados) que contaban con 2 transiciones les fueron asignadas etiquetas desde 1 hasta 7. Al resto de patrones (todos tienen más de 2 transiciones) les fue asignada la etiqueta 9, ver figura 3:

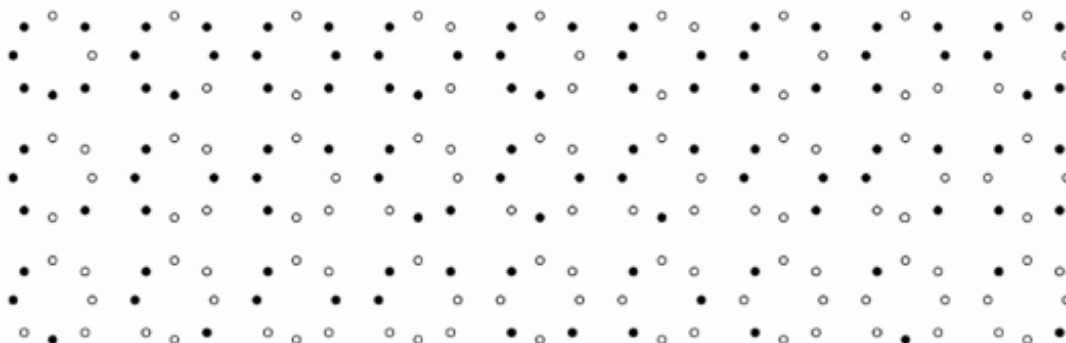


Figura 3.

En el fichero Datos.mat se proporciona un total de 1312 instancias, cada una de ellas corresponde a un bloque de tejido histopatológico de próstata con tamaño 2048x2048. De estas instancias, 1014 proceden de tejido sano (clase 0) y el resto, 298, de tejido cancerígeno (clase 1). **El problema está, por tanto, altamente desbalanceado.**

Con el objetivo de hacer una validación cruzada con 5 carpetas (*folds*), los bloques sanos y cancerígenos fueron distribuidos en 5 conjuntos disjuntos, que han sido guardados en las estructuras

Healthy\_folds y Malign\_folds, respectivamente. Haciendo doble clic en dichos objetos se debe ver la siguiente estructura:

Healthy_folds		Malign_folds	
1x5 struct with 1 field		1x5 struct with 1 field	
Fields	histogram	Fields	histogram
1	203x10 double	1	54x10 double
2	210x10 double	2	72x10 double
3	206x10 double	3	53x10 double
4	196x10 double	4	50x10 double
5	199x10 double	5	69x10 double
6		6	
7		7	
8		8	
9		9	
10		10	

La forma de acceder a los datos de cada partición  $i=1, \dots, 5$  es:

- `Healthy_folds(i).histogram`
- `Malign_folds(i).histogram`

Para un fold  $k$  ( $k=1, \dots, 5$ ), la forma de construir el conjunto de entrenamiento y test es la siguiente:

- El conjunto de test estará compuesto por los datos de `Healthy_folds(k).histogram` y `Malign_folds(k).histogram`.
- El de entrenamiento estará formado por los datos en `Healthy_folds(i).histogram` y `Malign_folds(i).histogram` para  $i$  distinto de  $k$  debidamente concatenados (uno bajo otro).

Muy importante, como el conjunto de entrenamiento está altamente desbalanceado (el número de casos negativos –no cáncer- es aproximadamente cuatro veces el de positivos –cáncer-), para cada *fold* tendrás que construir cuatro clasificadores GP que enfrenten las instancias positivas en un *fold* con cuatro subconjuntos disjuntos de las negativas de dicho *fold*. Estos cuatro subconjuntos serán de aproximadamente el mismo tamaño y deben cubrir todo el conjunto de instancias en el *fold*. Con un ejemplo, para fold 1, tus instancias de entrenamiento son: sanas (206, 210, 196, 199) y malignas (72, 53, 50, 69) y de test 203 (sanas) y 54 (malignas), respectivamente.

Comprueba que tus *folds* tienen las siguientes dimensiones:

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Tamaño-test	(203+54)x10	(210+72)x10	(206+53)x10	(196+50)x10	(199+69)x10
Tamaño-train	1055x10	1030x10	1053x10	1066x10	1044x10

Observa que cada partición, Fold1 a Fold 5, tiene, como cabría esperar, 1312 instancias.

Utiliza cualquier software que implemente el uso de GP para clasificación usando la función sigmoide para construir el clasificador. Los siguientes paquetes te pueden ser de utilidad:

- Para MATLAB (quizás el más antiguo, pero tiene los métodos básicos que necesitas): *GPML*. Enlace: <http://www.gaussianprocess.org/gpml/code/matlab/doc/>
- Para Python (sin TensorFlow, desarrollado por el grupo de Neil Lawrence): *GPy*. Enlace: <http://sheffieldml.github.io/GPy/>
- Para Python (con TensorFlow, el más potente, aunque también tiene los métodos básicos): *GPflow*. Enlace: <https://github.com/GPflow/GPflow>

Debes tener en cuenta que los rasgos no están normalizados, debes normalizarlos.

## Uso de GPML para realizar la práctica

Si usas el código en GPML en Matlab las siguientes líneas te serán de utilidad:

Como ya sabes, un GP es una colección de variables aleatorias, cada número finito de las cuales tienen una distribución conjunta Gaussiana. Para definir el proceso Gaussiano basta con especificar dicha distribución Gaussiana a través de la función media y covarianza. Estas funciones se especifican de forma separada. Habrá que indicar la forma de dichas funciones y los parámetros de los que dependen, llamados hiperparámetros, que deberán ser aprendidos.

- En el fichero `meanFunctions.m` podrás comprobar que hay numerosas funciones disponibles para la media. Habitualmente, se supone que la media es igual a cero, por lo que bastará con que definas `meanfunc=@meanZero`.
- En el fichero `covFunctions.m` se dispone de cantidad de funciones para definir la matriz de covarianza. Te puede ser de especial utilidad la función `@covSEiso` que conocemos como función de base radial o núcleo Gaussiano y que frecuentemente recibe el nombre de squared exponential o squared exponentiated. Tal y como vimos en clase, esta función depende de dos parámetros, habitualmente llamados varianza y escala, que en este paquete son denotados por `sf` y `ell`, respectivamente. Estos parámetros tienen que ser estimados durante la fase de aprendizaje (training). Los podemos inicializar usando `ell=1.9`, `sf=1.0` y definiendo `hyp.cov=log([ell sf])`.
- Con estos dos pasos hemos definido la distribución a priori, ahora definiremos el modelo de observación. Vamos a asignar la etiqueta  $y=1$  a tejido cancerígeno e  $y=-1$  a tejido no cancerígeno. Nuestro modelo de observación es

$$p(y|f) = \left(\frac{1}{1 + e^{-f}}\right)^{(1+y)/2} \left(\frac{1}{1 + e^f}\right)^{(1-y)/2}$$

que es la llamada función sigmoide o asociada a la regresión logística. Especificaremos en el modelo de observación que usamos `likfunc=@likLogistic`.

- Ya tenemos todos los ingredientes del problema. Tenemos el modelo a priori y el modelo de observación, es decir,  $p(y|f)p(f)$ . Recuerda que tenemos que estimar los parámetros del modelo  $\ell$  y  $sf$  que aparecen en  $p(f)$  resolviendo

$$\max_{\ell, sf} \int p(y|f)p(f)df$$

Fíjate que el modelo de observación no añade parámetros a estimar. El aprendizaje de parámetros se realiza con:

```
hyp= minimize(hyp, @gp, -40, @infVB, meanfunc, covfunc, likfunc, data, labels)
```

donde `data` será el conjunto de entrenamiento (instancias x características) aproximadamente balanceado y `labels` será un vector columna con las etiquetas, recuerda, etiqueta  $y= 1$  a tejido cancerígeno e  $y=-1$  a tejido no cancerígeno. La probabilidad de ser sana, `prob_test`, para cada instancia de test se obtendrá a partir de los parámetros `hyp` aprendidos de esta forma:

```
[a b c d lp]=gp(hyp, @infVB, meanfunc, covfunc, likfunc, data, labels, data_test, ones(n,1));
```

```
prob_test=exp(lp);
```

- Cuando quieras utilizar el núcleo lineal `@covLIN`, <https://github.com/trungngv/agp/blob/master/libs/gpml-matlab-v2/covLIN.m>, no tendrás que aprender (ni inicializar en consecuencia) los hiperparámetros  $sf$  y  $\ell$ , como ocurre con el gaussiano, <https://github.com/guruucsd/matlab-utils/blob/master/web/gpml/cov/covSEiso.m>, y bastará con establecer `hyp=[ ]`.
- Recuerda que con las particiones que has hecho en cada *fold*, el procedimiento necesitarás llevarlo a cabo cuatro veces por *fold*. Cuando hayas obtenido la probabilidad media de pertenecer a clase cancerígena para cada instancia de test en el *fold* correspondiente, deberás calcular

```
[X1, Y1, T, AUC1] = perfcurve(label_test, probability_mean,1);
```

y también

```
[X2, Y2, T2pr, AUC2] = perfcurve(targets, scores, 1, 'xCrit', 'sens', 'yCrit', 'prec');
```

Entiende que devuelven estas funciones (lo preguntaré en la defensa).

- A continuación, calcularás las predicciones (`label_pred_test`) fijando una threshold  $\theta$  entre 0 y 1. Utiliza  $\theta=0.5$  y si quieres incluye también otro valor que a ti te parezca más apropiado. Si la probabilidad para una instancia de ser cancerígena es menor o igual a  $\theta$ , clasificarás la instancia como sana (-1), en caso contrario, como cancerígena (1). Sobre estas predicciones calcularás la matriz de confusión con la función `confusionmat`:

`Confusion=confusionmat(label_test,label_pred_test)`

que devolverá la matriz en la forma

	Predicciones	
Etiquetas	TP	FN
	FP	TN

donde P representa a la clase positiva (en nuestro caso a la clase cáncer). Ten cuidado con la salida de confusionmat, que puede que no coincida con la anterior, tienes que tener claro qué es positivo (debe ser cáncer) y negativo (debe ser no cáncer). Puedes especificar el orden con la opción 'Order' <https://es.mathworks.com/help/stats/confusionmat.html>.

A partir de la matriz obtenida por la función confusionmat, calcula:

```
accuracy=(TP+TN)/(TP+FN+FP+TN)
specificity=TN/(TN+FP);
sensitivity (recall)=TP/(TP+FN);
precision=TP/(TP+FP);
F_score=(2*precision*sensitivity)/(precision+sensitivity);
```

## Documentación a entregar

Sube al servidor un documento en formato pdf. El nombre del fichero será tus apellidos y nombre seguidos de las iniciales GP. El fichero contendrá el título: GPCNombreApellido y las siguientes secciones:

- **¿Qué es un proceso gaussiano?** (media página, aproximadamente)  
En esta sección incluirás una breve descripción de los Procesos Gaussianos.
- **Software utilizado para la realización de la práctica** (media página, aproximadamente)  
En esta sección describirás qué paquete has usado para la realización de la práctica. Los núcleos utilizados y cualquier información relativa a como se ha realizado la estimación de los parámetros. Usa al menos dos núcleos: lineal y Gaussiano (SE).

- **Resultados experimentales.**
  - a. **Por *fold* debes hacer lo siguiente** (2 páginas, aproximadamente)
    - i. Para cada instancia de test en el *fold* debes haber obtenido cuatro probabilidades de que sea cancerígena (una por cada uno de los cuatro GPs). Calcula la media de estas cuatro probabilidades. A partir de ellas, dibuja la curva ROC e indica su área bajo la curva (AUC). Dibuja también la curva Precisión-Recall (PR) e indica también su área bajo la curva.
    - ii. Para un umbral 0.5 incluye la matriz de confusión y crea una tabla que contenga los valores de accuracy, specificity, sensitivity (recall), precision y F\_score.
    - iii. Discute los resultados obtenidos.
  - b. **Para un nuevo dato** (1 página, aproximadamente)
    - i. ¿Cómo lo clasificarías?
  - c. **Diseño de experimento adicional (no hay que implementarlo, solo discutirlo)** (1 página, aproximadamente)
    - i. En la práctica descrita hemos utilizado bagging para balancear las clases. Tenemos 1014 y 298 ejemplos sanos y cancerígenos, respectivamente. ¿Cómo aumentarías el número de ejemplos positivos para balancear los datos? Una vez balanceados, cómo construirías el clasificador.

La fecha límite de entrega es el 15 de marzo de 2019. La defensa de la práctica se realizará en el despacho del profesor a partir del día 18 de marzo de 2019 en horario que se publicará en la página web de la asignatura.