APR – Practica 2

Por Adrián Tendero Lara

# Resultados Sprinkler

En esta practica he realizado la actividad opcional del sprinker de realizar diferentes aprendizajes con distinto numero de muestras 100 y 1000 asi como también el experimento de aprendizaje con muestra oculta a 50% con los dos tamaños anteriores de muestra.

La red Bayesiana original redB se compone de 4 nodos (C,S,R,W) de los cuales el nodo inicial o fuente es el Nodo C que esta conectado a los nodos S y R y estos dos a su vez están conectados a W, cada nodo indica las posibilidades de que dado varios sucesos lo que representa el nodo sea cierto.

De tal modo que para redB tenemos las siguientes probabilidades:

P(W) = [1.0 0.1 0.1 0.01 0.0 0.9 0.9 0.99] --> [000,001,010,011,100,110,111] ->[SRW]

P(C) = [0.5 0.5] -> [0,1] -> [C]

P(S) = [0.5 0.9 0.5 0.1] ->[00,01,10,11] -> [CS]

P(R) = [0.8 0.2 0.2 0.8] -> [00,01,10,11] ->[CR]

De tal modo que el objetivo del aprendizaje es que la nueva redBayesiana obtenga unas probabilidades condicionales similares a la original mediante el entrenamiento, lo cual se obtiene los siguientes resultados:

Red Bayesiana producto del aprendizaje sin muestra oculta y con un numero de muestras = 100

P(W) = [1.0000 0.0556 0.0435 0.0000 0.0000 0.9444 0.9565 1.0000] --> [SRW]

P(C) = [0.47 0.53] -> [0,1] -> [C]

P(S) = [0.5532 0.9057 0.4468 0.0943] -> [CS]

P(R) = [0.7234 0.2264 0.2766 0.7736] -> [CR]

Red Bayesiana producto del aprendizaje sin muestra oculta y con un numero de muestras = 1000

P(W) = [1.0000 0.1085 0.0905 0.0250 0.0000 0.8915 0.9095 0.9750] --> [SRW]

P(C) = [0.4860 0.5140] -> [0,1] -> [C]

P(S) = [0.4979 0.9066 0.5021 0.0943] -> [CS]

P(R) = [0.8066 0.1673 0.1934 0.8327] -> [CR]

Red Bayesiana producto del aprendizaje con muestra oculta de 50% y con un numero de muestras = 100

P(W) = [1.000 0.0659 0.0136 0.1119 0.0000 0.9341 0.9864 0.8881] --> [SRW]

P(C) = [1.000 0.0000] -> [0,1] -> [C]

P(S) = [0.6673 1.000 0.3327 0.000] -> [CS]

P(R) = [0.4085 0.0000 0.5915 1.000] -> [CR]

Red Bayesiana producto del aprendizaje con muestra oculta de 50% y con un numero de muestras = 1000

P(W) = [0.9988 0.2952 0.1496 0.0601 0.0012 0.7048 0.8504 0.9399] --> [SRW]

P(C) = [0.9996 0.0004] -> [0,1] -> [C]

P(S) = [0.6969 1.000 0.3031 0.000] -> [CS]

P(R) = [0.4247 0.0000 0.5753 1.000] -> [CR]

Como se pueden observar en los resultados el aprendizaje que mas ha acertado es el aprendizaje realizado sin muestras ocultas y con mas muestras, en nuestro caso el que tiene Nmuestras = 1000.

Si se comprueba los resultados expuestos anteriormente se puede observar que los aprendizajes realizados sin muestra oculta han dado mejor resultado que los aprendizajes realizados con muestra oculta, es mas estos últimos han dado resultados totalmente erráticos en algunos casos como puede ser la probabilidad ha priori del nodo C, que en el original se encuentra a [0.5,0.5] y en ambos casos los resultados de los entrenamientos nos han dado un resultado muy parecido ha [1.0,0.0] en el resto de nodos los resultados son mas coherentes con la red original pero muestran una discrepancia mayor que los entrenamientos realizados sin muestra oculta.

# Ejercicio de Tasa de Error de clasificación

El ejercicio propone que realicemos una función que reciba como entrada en número de gaussianas por mixtura y los nombres de los ficheros, en mi caso he trabajado desde dentro de la carpeta de datos, aunque la ruta tendría que funcionar bien.

En mi caso he recabado resultados con diferente numero de gaussianas por mixtura (Ngauss) de un rango de 2 hasta 50 cabe comentar que el resultado con un Ngauss = 50 me ha dado problemas y el resultado puede no ser fiable.

Ha continuación voy a adjuntar mis resultados en una gráfica:

Como se puede observar la tasa de fallos se reduce constantemente ha partir de Ngaus de 10 o mas exceptuando para Ngaus mayores de 50 que causa problemas ha la hora de realizar los cálculos (indicado el error que salta en Ngaus 50 en el adjunto de resultados) también cabe mencionar que las semillas son aleatorias por lo que repitiendo el mismo calculo este podría generar resultados ligeramente mejores o peores pero con los resultados expuestos en la gráfica se puede observar la tendencia general de mantenerse sobre un 37% de tasa de fallos ha partir de Ngaus mayores que 10.

También cabe destacar la gran cantidad de tiempo consumido en todos los cálculos incrementando de manera bastante notable para Ngaus grandes.

En conclusión, contra mas grande sea el numero de gaussianas por mixtura mayor es la precisión en la clasificación de los elementos de prueba pero ha costa de un gran tiempo de calculo, mientras que Ngauss menores de 10 generan unos resultados bastante peores que Ngaus mas elevadas pero a un coste temporal bastante menor.

# Notas

Aunque he realizado las pruebas de Gauss mediante una función contenida en un fichero m las pruebas con el Sprinker las he realizado ha mano en la consola, la lista de comandos que he empleado están contenidos en el fichero de texto “Ejemplo Sprinker’ y los resultados guardados en el fichero de texto “Resultados Sprinker”, en cuanto al ejercicio de gaussianas se encuentran los comandos en el fichero “Mixtura Gaussiana”, el fichero que contiene la función de tasa de error en el fichero “TasaError.m” cuya función se compone de 5 argumentos, se llama con la siguiente instrucción, “TasaError(FicheroEntrenamiento, Fichero de labels de entrenamiento, fichero de prueba, fichero con los labels de pruebas, Ngaus)” dentro de dicho fichero esta escrita una llamada de ejemplo ha esta función y por ultimo los resultados de la tasa de error se encuentra en el fichero de texto “REsultados Tasa Error”.

Por ultimo también quiero comentarle que tengo un repositorio de github donde se encuentra todos los ficheros de la practica (sin boletines) por si hay algún problema con la entrega del poliformat, como que falte algún archivo o algo.

[Repositorio GitHub de APR](https://github.com/Hollex28/APR)