DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico

Polynomial Regression

Reconocimiento de patrones

Integrante	LU	Correo electrónico
Rodrigo Oscar Kapobel	695/12	rokapobel35@gmail.com

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2160 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (54 11) 4576-3359
http://www.fcen.uba.ar

Índice

1. Introducción **3**
1.1. K-Means Clustering 3
1.2. Expectation-Maximization 4

2. Implementación **5**

3. Casos de estudio **6**
3.1. K-Means Clustering 6
3.2. Expectation-Maximization 6

4. Conclusiones **7**

1. Introducción

En este trabajo práctico se implementan los algorítmos de K-Means Clustering y Expectation-Maximization.

1.1. K-Means Clustering

La idea del algorítmo es tener, para cada clase k: $1...K$, una media μ_k de los datos de las mismas. Estas se actualizan iterativamente mediante la siguiente fórmula:

Para cada x_n, este será clasificado en la clase k que minimice el cuadrado de la norma euclidiana entre x_n y μ_k, Luego, cada μ_k es actualizado según:

$$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$

Donde r_{nk} será:

$$r_{nk} = \begin{cases} 1 & \text{si } k = argmin_j ||x_n - \mu_j||^2 \\ 0 & sino \end{cases}$$

El proceso de primero actualizar r_{nk} y luego μ_k se conoce respectivamente como Expectation y Maximization. Podemos definir la función objetivo de este algorítmo de la siguiente manera:

$$J = \sum_n^N \sum_k^K r_{nk} ||x_n - \mu_j||^2$$

La idea es encontrar los valores de r_{nk} y μ_k que minimizan J.

La convergencia se puede determinar por cantidad de iteraciones. Este hiperparámetro necesita ser ajustado dependiendo del set de datos. Para este informe los datos son generados a partir de las mismas distribuciones en cada ejecución y será fijado en 200 iteraciones ya que se han obtenido buenos resultados en la mayoría de los casos. Podrá modificarse en los tests ya que es un parámetro de K-Means.

Este algorítmo no depende solamente del hiperparámetro cantidad de iteraciones, si no que también depende fuertemente de la inicialización de las medias. Para este informe se elige un subset de K puntos de los datos de training.

Esta implementación del algorítmo K-means puede ser relativamente lenta, dado que en cada paso de expectación es necesario calcular la norma euclidiana entre cada media y cada dato. Varios esquemas se han propuesto para acelerar el algoritmo K-means, algunos de los cuales se basan en precomputar una estructura de datos tal como un árbol en el que los puntos cercanos están en el mismo subárbol (Ramasubramanian y Paliwal, 1990; Moore, 2000).

Otros enfoques hacen uso de la desigualdad triangular para las distancias, evitando así cálculos innecesarios (Hodgson, 1998; Elkan, 2003).

Se ha considerado una versión por clusters de K-means en la que toda la información se usa en conjunto para actualizar μ_k. También puede derivarse un algorítmo estocástico (MacQueen, 1967) mediante la aplicación del procedimiento de Robbins-Monro al problema de encontrar las raíces de la función de regresión dada por las derivadas de la función objetivo J con respecto a μ_k. Esto lleva a una actualización secuencial en la cual, para cada x_n actualizamos la media más cercana usando:

$$\mu_k^{\tau+1} = \mu_k^{\tau} + \eta_n (x_n - \mu_k^{\tau})$$

Donde η_n es el parámetro de learning rate que está considerado para decrecer monótonamente a medida que más puntos son considerados como pertenecientes a la clase k.

1.2. Expectation-Maximization

La distribución de mixturas de Gaussianas se puede expresar como la superposición lineal de Gaussianas:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\mu_k, \Sigma_k)$$

Donde π_k se define como el coeficiente de mixtura y satisface:

$$0 \leq \pi_k \leq 1$$

$$\sum_{k=1}^{K} \pi_k = 1$$

Si definimos $\boldsymbol{z}$ como el vector 1-K tal que $p(z_k = 1) = \pi_k$, luego la probabilidad de $\boldsymbol{z}$ se define como:

$$p(\boldsymbol{z}) = \prod_{k=1}^{K} \pi_k^{z_k}$$

De manera similar, la probabilidad condicional de $\boldsymbol{x}$ dado $\boldsymbol{z}$ es:

$$p(\boldsymbol{x}|\boldsymbol{z}) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{x}|\mu_k, \Sigma_k)^{z_k}$$

De esta manera se obtiene, por Bayes:

$$p(\boldsymbol{x}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z}) = \sum_{\boldsymbol{z}} p(\boldsymbol{z}) p(\boldsymbol{x}|\boldsymbol{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\mu_k, \Sigma_k)$$

Por lo tanto, cada dato $\boldsymbol{x}_n$ tendrá una variable latente $\boldsymbol{z}_n$.

Otra medida que es importante es la denominada responsabilidad definida como la probabilidad condicional de $\boldsymbol{z}$ dado $\boldsymbol{x}$ y puede hallarse usando Bayes:

$$\gamma(z_k) = p(z_k = 1|x) = \frac{p(z_k=1)p(\boldsymbol{x}|z_k=1)}{\sum_j p(z_j=1)p(\boldsymbol{x}|z_j=1)} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}|\mu_k,\Sigma_k)}{\sum_j \pi_j \mathcal{N}(\boldsymbol{x}|\mu_j,\Sigma_j)}$$

Luego, el algorítmo de Expectation-Maximization se define en los siguientes pasos.

1. Inicializar μ_k, Σ_k y π_k y evaluar la log verosimilitud inicial.
2. Expectation-Step: Actualizar las responsabilidades $\gamma(z_k)$
3. Maximization-Step: Re-estimar μ_k, Σ_k y π_k como:
   a) $\mu_k^{\tau+1} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{x}$
   b) $\Sigma_k^{\tau+1} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\boldsymbol{x} - \mu_k^{\tau+1})(\boldsymbol{x} - \mu_k^{\tau+1})^t$
   c) $\pi_k^{\tau+1}) = \frac{N_k}{N}$
4. Actualizar la log verosimilitud:

   $$\boldsymbol{ln}\ p(\boldsymbol{X}|\Sigma, \mu, \pi) = \sum_{n=1}^{N} \boldsymbol{ln} \left\{ \sum_{k=1}^{K} \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k) \right\}$$

La justificación de las fórmulas planteadas en los pasos 3 y 4 son bien desarrolladas en el libro Pattern Recognition and Machine Learning de Bishop sección 9.2.2 EM for Gaussian Mixtures.

El objetivo de EM es maximizar la log verosimilitud. Para determinar la convergencia, en este informe se utiliza, cantidad de iteraciones máxima en 2000 iteraciones. Este hiperparámetro puede ser modificado en los tests dado que es un parámetro de EM.

Hay que tener en cuenta que el algoritmo EM requiere muchas más iteraciones para alcanzar (aproximadas) convergencia en comparación con K-Means (Para los tests, 2000 vs. 200 iteraciones respectivamente) y que cada ciclo requiere significativamente más computo. Además de que depende de la inicialización al igual que K-Means. Por lo tanto, es común ejecutar K-Means para encontrar una inicialización adecuada para EM.

Para este informe, se elige inicializar estos parámetros independientemente de K-Means, pero realizando un proceso como el de este último. Se toman las medias de la misma manera que en K-Means. Las covarianzas serán las identidades $D \times D$ con D la dimensión de los datos. Los coeficientes de mixtura serán $\frac{1}{K}$.

Se debe enfatizar que generalmente habrá múltiples máximos locales para la log verosimilitud y que por lo tanto EM no está garantizado a encontrar el mayor de estos máximos.

2. Implementación

El test corre para un mismo set de datos de entrenamiento, K-Means y EM. De esta manera podrá validar su funcionamiento para un mismo set de datos. Debe ingresar el siguiente comando en el terminal desde el directorio TP3:

```
python GaussianMixturesTest.py
```

para más opciones:

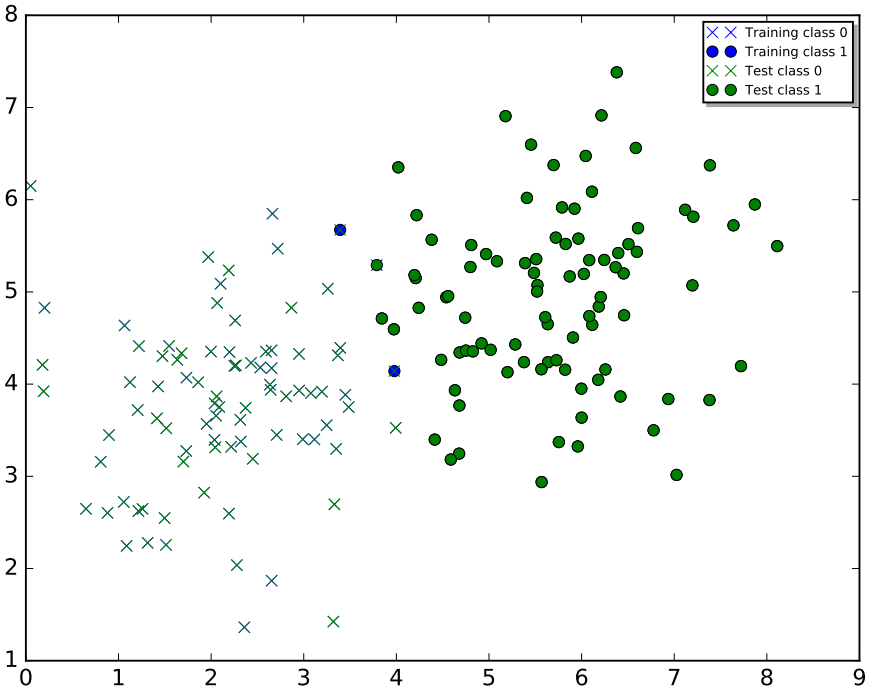
```
python PolynomialRegressionTest.py -h
```

Podrá elegirse la cantidad de clases, el número de iteracioens para cada algorítmo y que algorítmos correr ingresando los comandos indicados por help.

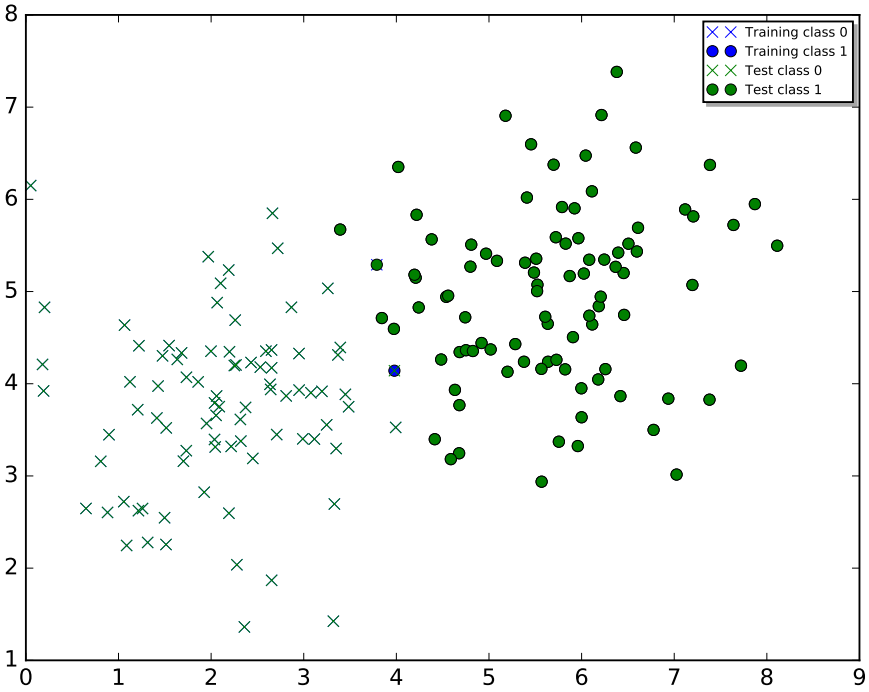
```
-k NUMBEROFCLASSES  Number of classses.
-t TESTSTORUN
0: run EM and K-Means.
1: Will run only EM.
2: Will run only K-Means.
-em NUMBEROFITERATIONSOFEM
Number of iterations of EM.
-km NUMBEROFITERATIONSOFKMEANS
Number of iterations of K-Means.
```

3. Casos de estudio

3.1. K-Means Clustering



3.2. Expectation-Maximization



4. Conclusiones

1. A comparación de K-Means, EM requiere realmente muchas más iteracioens y más cómputo para obtener los mismos resultados.

2. Los hiperparámetros y las inicializaciones de los parámetros de cada método fueron difíciles de ajustar. En muchos casos EM no llega a una convergencia adecuada y es posible que se deba al método de inicialización elegido a pesar de lo aclarado en el punto anterior.