Facultad de Ingeniería de la
Universidad de la República

Introducción al Reconocimiento de Patrones

Curso 2013

Proyecto final

Autores:
José Luis Nunes
Matías Tailanián

Tutores:
Federico Lecumberry
Ignacio Ramirez

12 de diciembre de 2013

Índice general

Introducción

El objetivo principal del proyecto es la investigación en técnicas que permitan contribuir con la predicción de fertilidad de rodeo lechero y la calidad de la carne integrando técnicas de reconocimiento de patrones sobre datos de alta dimensión.

A lo largo de los años se ha intentado relacionar ciertas características fenotípicas del ganado bovino con algunos indicadores genéticos. Es un problema actualmente abierto de gran interés mundial y Uruguay, siendo un país esencialmente ganadero, no puede estar ajeno. Desde hace varios años se viene desarrollando una línea de investigación relacionada con la genética molecular de la calidad de la carne bovina en el área de Genética de la Facultad de Veterinaria. Se ha realizado un abordaje desde diferentes puntos de vista, como la caracterización de genes conocidos en rodeos vacunos de nuestro país, o la búsqueda y análisis de nuevos genes asociados a la calidad de carne tanto bovina como ovina [1].

Obtener una correlación entre la información genética y la capacidad reprodutiva de bovinos, el impacto en la calidad de la leche y de la carne de los animales de nuestros rodeos tendría un alto impacto en la producción. A lo largo de los últimos años en Facultad de Veterinaria se han creado bases de datos con información fenotípica y genotípica relacionada conla calidad cárnica y la fertilidad bovina aplicada a la producción lechera. En esta oportunidad se trabajará con una base de datos confeccionada en nuestro país durante el año 2009, provista por la veterinaria Dra. Ana Meikle.

Base de datos

Para lo que sigue de este trabajo es importante conocer la base de datos con la que se trabajará, y realizar una breve descripción de las características relevantes.

La base de datos consta de información de un seguimiento realizado durante 9 meses sobre 891 vacas de 7 tambos diferentes y consiste en las características detalladas a continuación.

2.1. Características fenotípicas

- Edad.
- Condición corporal (BCS) desde 30 días previos al parto hasta 120 días posparto, con una frecuencia de al menos una vez cada 30 días. Se utilizó la escala de 5 puntos (1=flaca, 5= gorda).
- Cantidad de partos (Cantidad de lactancias). Es un indicador importante que tiene que ver con la historia y desgaste de la vaca.
- Anestro: es la cantidad de días que pasaron desde el parto hasta el reinicio. El reinicio se define como el día en que la progesterona aumenta a un nivel determinado, indicando que la vaca volvió a ciclar.
- Intervalo entre partos. Es la cantidad de días que pasan entre 2 partos consecutivos (válido solamente para vacas multíparas). Para maximizar la producción de leche lo que se busca es que la vaca quede peñada una vez al año, es decir un intervalo entre partos de 365 aproximadamente.
- Secado: la cantidad de días que pasan entre el último día que se le saca leche a la vaca y el parto. Es válido solo para vacas multíparas. Es un indicador de cuanto tiempo se dejó descansar a la vaca antes del parto. Cuanto más tiempo tiene, se prepara el cuerpo y llega en mejor forma.
- Servicios: Cantidad de inceminaciones realizadas para lograr la preñez.
- Concentración de progesterona hasta los 60 días posparto.

- Promedio de cantidad de grasa en la leche durante los primeros 100 días posparto.
- Promedio de cantidad de leche durante los primeros 100 días posparto.

2.2. Características genotípicas - Determinación de polimorfismos (SNPs)

Se extrajo ADN y se obtuvieron muestras de buena calidad. En la figura 2.1 se muestra el procedimiento realizado para caracterizar el gen IGF-I bovino en 3 clases: "AA", "AB" y "BB".

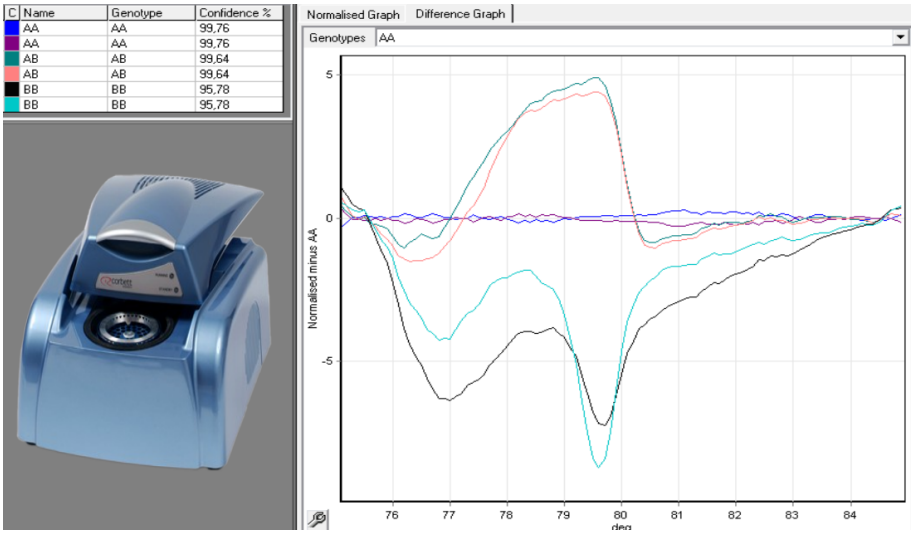


Figura 2.1: Determinación del genotipo

El eje de las ordenadas muestra la fluorescencia normalizada al genotipo "AA". Se puede observar que el genotipo "AB" se encuentra por encima del genotipo normalizado, mientras que el genotipo "BB" por debajo.

2.3. Resumen

En primer lugar fue necesario entender e interiorizarse con la base de datos, los conceptos y la terminología específica del problema. La lectura e interpretación de los datos no resulta para nada sencilla, por lo que se realizaron dos reuniones con la Dr. Ana Meikle, encargada de la investigación que llevó a cabo la confección de la base asociada a la producción de leche y fertilidad del ganado bovino. En ambas reuniones se buscó depurar la base con el fin de quitar información repetida y no relevante para el estudio. También se estableció un orden de relevancia en las características fenotípicas.

En resumen contamos con una base de datos acotada y "limpia" con varias características fenotípicas que se quieren correlacionar con los genotipos de cada individuo. En particular se abordará el problema como un trabajo de clasificación, tomando los genotipos como clases.

Técnicas utilizadas

La mayoría de los algoritmos de *machine learning* seleccionan los atributos apropiados para realizar sus decisiones. Por ejemplo los métodos de árbol de desición eligen en cada paso la característica que mejor separa en clases. Cuantas más características tengamos, en teoría tendremos más poder de discriminación, pero en la práctica no ocurre así. Agregar información irrelevante o características distractivas, confunde a los algorimos de machine learning. Está probado que el número de instancias necesarias para entrenamiento para producir un cierto nivel de performance crece exponencialmente con la cantidad de atributos irrelevantes.

Dado el efecto negativo de las características irrelevantes es muy común en muchos algoritmos de *machine learning* que se realice una etapa de selección de características donde se eliminen las irrelevantes. La mejor manera de realizar una selección de características es manual, adquiriendo un amplio conocimiento del problema en cuestión y logrando interpretar cada una de las características. Sin embargo es posible realizar una selección con algoritmos automáticos que resultan muy útiles. Reducir la dimensionalidad de los datos puede mejorar la performance de los algoritmos, además de reducir la complejidad del problema y bajar los requerimientos de capacidad de cómputo. Por esta razón se realizará una etapa de selección automática de características previa a la clasificación, como se explicará más adelante.

En una primera etapa de análisis sobre la base de datos se ataca el problema con los clasificadores **C4.5**, **Naive Bayes** y **K-NN**.

El algoritmo **C4.5** se utiliza para generar un árbol de desición que puede ser utilizado para clasificación. Por la naturaleza de los árboles, es un algoritmo no paramétrico, por lo que resulta robusto ante *outliers*.

Naive Bayes es un clasificador basado en la aplicación del teorema de Bayes. No particiona el espacio de instancias e ignora de forma robusta a las características irrelevantes. Asume por diseño que todas las características son independientes entre si, y paga un precio muy alto cuando hay características redundantes.

El algoritmo **k-NN** (k-nearest neighbors) es un algoritmo de clasificación super-

vizado no paramétrico que pedice la clase de las instancias basado en los k vecinos más cercanos. Es un algoritmo muy sensible a la propia estructura de los datos. Cuando $k \to \infty$, el algoritmo asegura una tasa de error no superior al doble de la tasa de error de Bayes (mínimo alcanzable dada la distribución de los datos).

Con esta batería de clasificadores se cubre un amplio espectro y se utilizan algunos de los algoritmos más utilizados para problemas de reconocimiento de patrones.

Para analizar los resultados se utilizan las implementaciones provistan en el software **Weka** [2].

3.1. Primera etapa

Se utiliza el clasificador compuesto AttributeSelectedClassifier que aplica una técnica de selección de características antes de entrenar al clasificador. Se logra entonces una reducción de la dimensionalidad. La estrategia elegida para la selección de características se realiza utilizando el enfoque wrapper , que evalúa el set de atributos utilizando un esquema de aprendizaje y utiliza validación cruzada para estimar la precisión del esquema de aprendizaje. El clasificador utilizado para estimar esta precisión es un árbol de decisión **C 4.5**.

En todos los casos se utilizó validación cruzada con 10 subconjuntos.

Una vez hecha la extracción de características, y así la reducción de dimensionalidad, se estudia el desempeño de diferentes clasificadores. En la tabla 3.1 se muestran los resultados para los algoritmos C4.5, Naive Bayes y k-NN.

	Tiempo [s]	$\sqrt{\textbf{MSE}}$	**F-Measure**	$\boldsymbol{\kappa}$	**Bien Clasif [%]**
C4.5	0.84	0.45	0.331	0	49.83
Bayes	0.37	0.45	0.345	0.0044	49.60
k-NN	0.37	0.53	0.402	0.0043	45.23

Tabla 3.1: Resultados etapa 1

Se puede observar en la tabla anterior que se obtienen resultados muy similares para los 3 algoritmos. Para C4.5 y Bayes se obtiene un porcentaje de aciertos un poco menor al 50 %, mientras que para k-NN los resultados son un poco inferiores.

La medida de error *kappa-statistic* ($\boldsymbol{\kappa}$), es un indicador de la performance del algoritmo que tiene en cuenta las coincidencias por azar. Se calcula como

$$\boldsymbol{\kappa} = \frac{P_0 - P_e}{1 - P_e}$$

donde P_0 es la proporción de coincidencias observadas y P_e la proporción de coincidencias esperadas en las hipótesis de independencia, es decir, coincidencias por azar. Se puede ver en la tabla 3.1 que se obtuvieron valores de $\boldsymbol{\kappa}$ realmente bajísimos, siendo este un indicador más de la mala performance alcanzada por los algoritmos.

Por otro lado resulta interesante analizar las matrices de confusión que resultan de estos algoritmos. Para el C4.5 se obtiene la siguiente matriz:

```
1     a   b   c   <-- classified as                                    C4.5
2     0 309   0 |   a = AA
3     0 444   0 |   b = AB
4     0 138   0 |   c = BB
```

Claramente el resultado obtenido no es el esperado. En este caso clasifica todos los patrones como pertenecientes a la clase "AB", y como esta clase representa casi el 50 % de todas las muestras, el porcentaje de aciertos coincide. Este es un resultado determinístico, que más allá del porcentaje de aciertos, significa que el clasificador no funcionó adecuadamente. A su vez, analizando el árbol de decisión se puede ver que tiene una sola hoja.

Por otro lado las matrices de confusión para los algoritmos *Naive Bayes* y *k-NN* son las siguientes:

```
1     a   b   c   <-- classified as                                   Bayes
2     6 300   3 |   a = AA
3     8 435   1 |   b = AB
4     1 136   1 |   c = BB
```

```
1     a   b   c   <-- classified as                                    k-NN
2    63 233  13 |   a = AA
3    81 331  32 |   b = AB
4    18 111   9 |   c = BB
```

Si bien estos dos casos no se obtuvo un resultado determinístico como con C4.5, igualmente los resultados tienen un fuerte sesgo hacia la clasificación de los patrones como pertenecietes a la clase "AB".

Para mitigar el fenómeno de la salida determinística (y el sesgo) mencionado, el siguiente paso es atacar el problema del desbalance de clases. Para ello se realiza un sorteo aleatorio de las muestras pertenecientes a las 2 clases mayoritarias, de forma que las 3 clases tengan la misma cantidad de patrones. Los resultados se presentan en la siguiente sección.

3.2. Segunda etapa - Clases balanceadas

Las características seleccionadas son la condición corporal al momento del parto, a los 30, 45 y 90 días post parto, la cantidad de lactancias, la edad, el intervalo entre partos, progesterona y la cantidad de leche.

Los resultados de la segunda etapa, balanceando las clases, se muestran en la figura 3.2.

	Tiempo [s]	$\sqrt{\text{MSE}}$	F-Measure	κ	Bien Clasif [%]
C4.5	12.57	0.54	0.39	0.080	38.65
Bayes	11.77	0.50	0.30	0.044	36.23
k-NN	11.92	0.63	0.38	0.069	37.92

Tabla 3.2: Resultados etapa 2

Como primer ítem a mencionar se debe destacar el descenso en el porcentaje de aciertos, de algo más del 10 %. Mientas que los resultados de la primera etapa arrojaban un porcentaje de aciertos de aproximadamente 49 %, en esta etapa se nota un descenso hasta alrededor de los 38 puntos pocentuales. Aunque a priori parece un peor resultado, en interesante analizarlo con cuidado ya que por ejemplo el índice κ aumentó un órden de magnitud, aunque sigue siendo muy malo.

A diferencia de los resultados de la etapa anterior, para el caso del algoritmo C4.5 se obtiene un árbol no trivial con una cantidad total de 147 nodos y 74 hojas. Si bien la clasificación se encuentra por encima de una una clasificación aleatoria, son resultados realmente muy malos.

El balanceo de clases logró solucionar el problema del resultado determinístico (o el fuerte sesgo) donde se clasificaba todos los patrones (o casi todos) como pertenecientes a la clase "AB", y la matriz de confusión se muestra a continuación:

```
1   a   b   c   <-- classified as                    C4.5
2  44  55  39 |  a = AA
3  51  54  33 |  b = AB
4  40  36  62 |  c = BB
```

```
1   a   b   c   <-- classified as                    Bayes
2  10 102  26 |   a = AA
3   6 107  25 |   b = AB
4  19  86  33 |   c = BB
```

```
1   a   b   c   <-- classified as                    k-NN
2  45  52  41 |  a = AA
3  54  49  35 |  b = AB
4  31  44  63 |  c = BB
```

Dado el terriblemente bajo porcentaje de aciertos obtenido en esta etapa, en la siguiente etapa se intentará realizar una extracción de características con métodos más sofisticados e intentar explicar el por qué de los resultados tan malos.

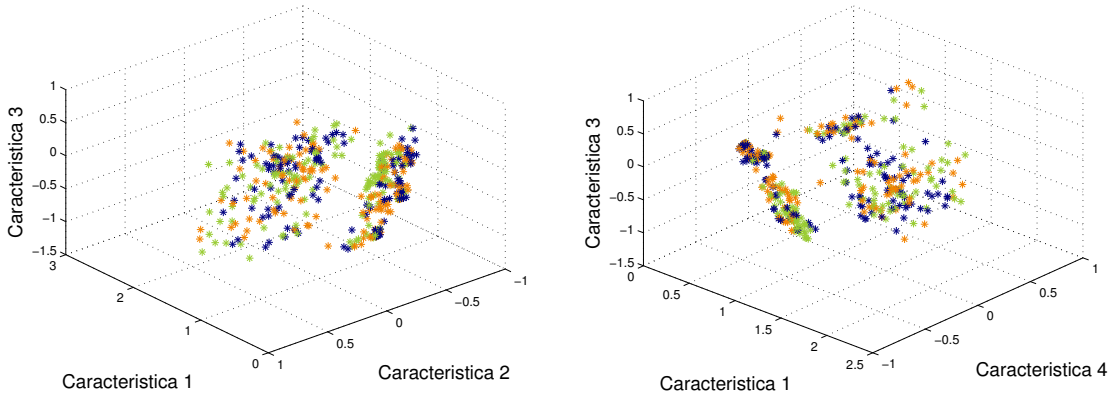
3.3. Tercera etapa - extración de características

En la tercera etapa buscaremos realizar extracción de características con el fin de reducir la dimensionalidad y buscar características con mayor discriminación.

Esto tiene como fin reducir los niveles de redundancia entre las características, visualizar características latentes significativas y generar para el futuro una mayor compresión en el proceso de generación de datos.

3.3.1. PCA

El algoritmo **PCA** (Análisis de componentes principales) tiene como fin encontrar la base de vectores que mejor exprese la distribución de los datos en el espacio completo. Es similar a encontrar las componentes ortogonales de un vector en un espacio, o lo que es igual, encontrar un conjunto de vectores que combinados en forma lineal representen los elementos. Estos elementos son los vectores propios de la matriz de covarianza correspondiente al espacio original. **PCA** tiene como fin encontrar un subespacio principal en el cual se maximice la varianza de los datos proyectados. Se ordenan las variables de acuerdo a la cantidad de varianza que concentran y se utilizan solamente las más significativas.



(a) Características 1-2-3 (b) Características 1-4-3

Figura 3.1: Datos procesados con el algorítmo PCA

A simple vista resulta muy difícil reconocer algún tipo de estructura sobre los datos, con lo cual es de esperar que la clasificación no entregue mejores resultados de los ya vistos. La distribución de los datos en ambos subespacios (ver figura **??**) resulta prácticamente randómica y es imposible identificar visualmente algún cluster por clases. (Esto es medio trucho, pero sincero...)

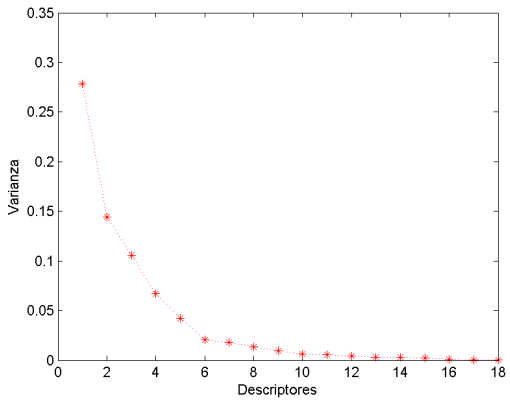


Figura 3.2: Varianza vs Componentes

En la figura 3.2 vemos la varianza en función de los componentes, podemos apreciar como la caída es abrupta y tiene sentido trabajar en el espacio de los primeros tres componentes que acumulan la mayor cantidad de varianza.

Los resultados de aplicar los clasificadores a los datos procesados con PCA se muestran en la tabla 3.3.

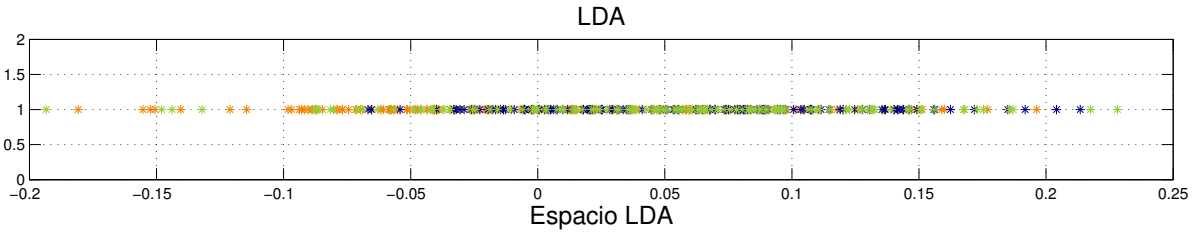


Figura 3.3: Proyección de los datos aplicando LDA

	Tiempo [s]	$\boldsymbol{\kappa}$	**Bien Clasif** [%]
C4.5	0.03	0.143	42.75
Bayes	0.02	0.091	39.37
k-NN	0	0.149	43.24

Tabla 3.3: Resultados etapa 2

Las matrices de confusión para cada clasificador son:

```
1    a   b   c   <-- classified as                    C4.5
2   50  42  46 |  a = 1
3   43  59  36 |  b = 2
4   39  31  68 |  c = 3
```

```
1    a   b   c   <-- classified as                    Bayes
2   19  93  26 |  a = 1
3   22  85  31 |  b = 2
4   18  61  59 |  c = 3
```

```
1    a   b   c   <-- classified as                    k-NN
2   58  45  35 |  a = 1
3   44  59  35 |  b = 2
4   44  32  62 |  c = 3
```

Como era de esperarse analizando la distribución de los datos, se obtuvieron resultados muy malos, aunque sensiblemente mejores que en la etapa anterior, llegando en el caso de *k-NN* a un porcentaje de aciertos de un poco más del 43 %.

A su vez, la medida estadística $\boldsymbol{\kappa}$ aumentó un orden de magnitud respecto a la etapa anterior, y dos órdenes respecto a la etapa 1.

3.3.2. LDA

El algorítmo LDA (Análisis de discriminantes lineales) tiene como fin seleccionar una proyección que maximice separabilidad inter-clases. Busca una proyección de los datos en un espacio de menor (o igual) dimensión que las iniciales con el fin de que la disciminabilidad inter-clases sea lo más alta posible. Es una técnica supervisada ya que para poder buscar dicha proyección se debe entrenar el sistema con patrones etiquetados.

En la figura 3.3 se muestran los resultados de aplicar LDA a los datos. Nuevamente resulta imposible obtener algun resultado visualizando la distribución de los datos, pero de todas formas se intenta realizar una clasificación con los algoritmos C4.5, Naive Bayes y k-NN. Estos resultados se muestran en la tabla 3.4

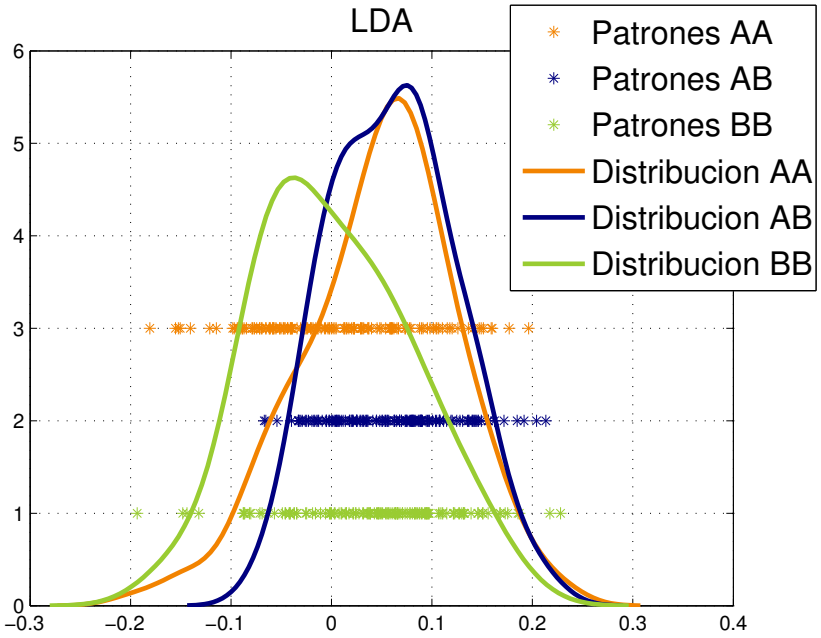


Figura 3.4: Estimación de la distribución de los datos

	Tiempo [s]	κ	**Bien Clasif [%]**
C4.5	0.03	0.1558	43.7198
Bayes	0	0.1667	44.4444
k-NN	0	0.1196	41.3043

Tabla 3.4: Resultados etapa 2

Como era de esperarse al analizar los patrones en el espacio transformado, nuevamente no es posible realizar una buena clasificación con ninguno de los clasificadores probados.

Las matrices de confusión para cada clasificador son:

```
1    a   b   c   <-- classified as                    C4.5
2    7 107  24 |   a = 1
3   12 122   4 |   b = 2
4    5  81  52 |   c = 3
```

```
1    a   b   c   <-- classified as                    Bayes
2   10  85  43 |   a = 1
3    7  94  37 |   b = 2
4    3  55  80 |   c = 3
```

```
1    a   b   c   <-- classified as                    k-NN
2   64  48  26 |   a = 1
3   66  54  18 |   b = 2
4   43  42  53 |   c = 3
```

Para corroborar de otra forma que los datos son efectivamente inconcluyentes se presenta la figura 3.4, donde se puede ver con asteriscos los patrones representados en el espacio LDA y con líneas sólidas las distribuciones estimadas por clase. La representación de los patrones se realizó utilizando un color y un nivel distinto para cada clase para ayudar a la claridad de visualización. Se puede corroborar una vez

más que las 3 clases son muy dificiles de separar, ya que presentan distribuciones realmente muy similares.

3.3.3. Diffusion Maps

Con el objetivo de probar alguna técnica más sofisticada se utiliza Diffusion Map [4].

Diffusion maps es un algoritmo de *machine learning* que computa una familia de conjuntos de datos en un espacio embebido, usualmente de baja dimensión, cuyas coordenadas pueden ser calculadas de los vectores y valores propios de un operador de difusión de los datos. La distancia euclídea entre puntos en el espacio embebido es la "distancia de difusión". A diferencia de otros métodos de reducción de dimensionalidad como PCA, este algoritmo es un método no lineal que se centra en descubrir *manifold* subyacente al muestreo de los datos. Integrando la similitud local de los datos a diferentes escalas, *diffusion maps* da una descripción global de los datos. Es robusto ante perturbaciones ruidosas y computacionalmente barato.

Los resultados de pasar a un espacio de 3 dimensiones (para poder visualizarlo) se muestran en la figura 3.5.

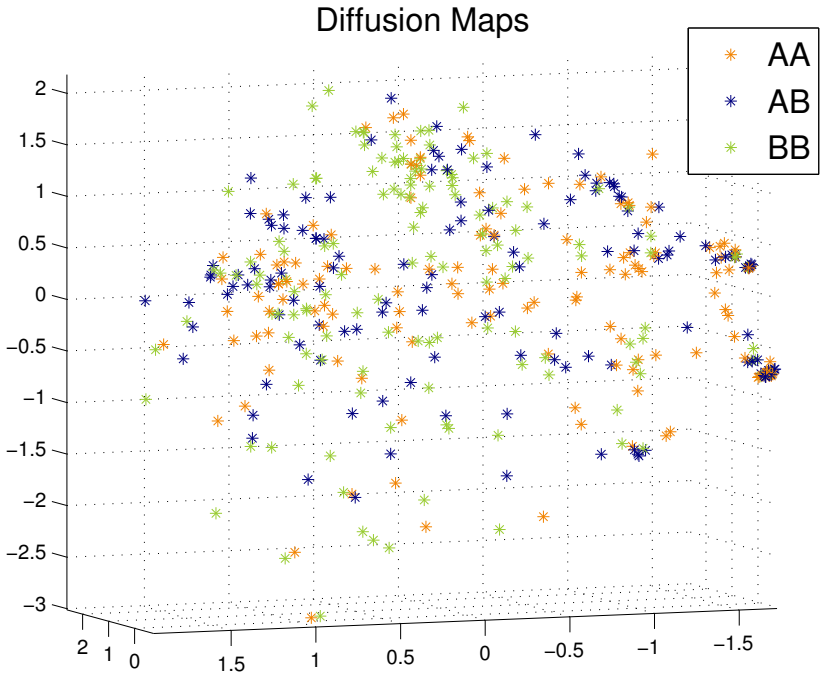


Figura 3.5: Diffusion Maps

Como se puede apreciar en la figura, al igual que en los casos anteriores los patrones de las diferentes clases están realmente muy entreverados, no pudiéndose diferenciar ningún cluster.

Conclusiones

Cuestionarse si un determinado gen puede tener efectos en variables fenotípicas resulta una pregunta ambiciosa incluso para un genetista, y por esa razón es que desde un principio tenemos presente la dificultad del problema al que nos enfrentamos. A diferencia de otro tipo de estudios donde uno trata de elegír descriptores que sospecha que pueden tener una realación con la clasificacón buscada, en nuestro caso buscabamos encontrar la relación entre un tipo de gen y sus marcadores moleculares respecto de un conjunto de variables fenotípicas, sin ningún tipo de conocimiento a priori. En este estudio nuestra parte consistía unicamente en realizar el análisis sobre los valores proporcionados. No tuvimos ninguna participación, como resulta obvio, en el diseño del experimento y la adquisición de datos. Es por eso que llegar a comprender el problema, las variables, los cuestionamientos e inclusive los objetivos, fue una tarea a la cuál se le debió dedicar un tiempo para nada despreciable dentro de los tiempos acotados del proyecto.

El problema fue atacado en su mayoria por las herramientas dadas en el curso, utilizando mayoritariamente el software weka y matlab. Se experimentó con una técnica un tanto más sofisticada como lo es "Diffusion Maps". A su vez, dado que este trabajo excede los objetivos de este curso ya que forma parte del proyecto de investigación en el que trabajamos, se intentará utilizar otro método para la clasificación: **Restricted maximum likelihood (REML)**, mediante la implementación en el software *Wombat* [3]. Es un método prometedor que es muy utilizado en problemas de este tipo.

Los resultados fueron contundentes, en la primera etapa, en la cual se trabajo con la base ya depurada pero sin ningun tipo de ajuste, los resultados no fueron los esperados. Todos los clasificadores repondieron en la misma manera comentiendo el mismo error, clasificando a practicamente la totalidad de los patrones en una misma categoría: "AB". Observando la cantidad de patrones por clase, como se muestra en la figura 4.1, destaca la categoría "AB" superando ampliamente a las restantes. Es probable que eso lleve a que los clasificadores obtengan el mejor resultado (aproximadamente 50 %) clasificando todos los patrones como "AB".

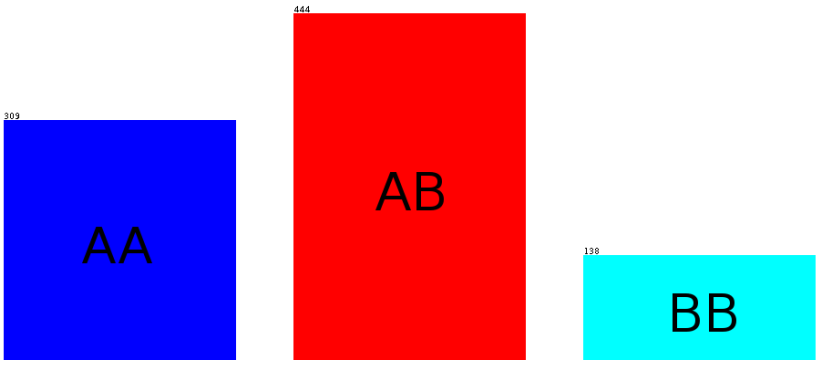


Figura 4.1

El siguiente paso fue trabajar o “masajear” la base de datos con el fin de balancear las clases y normalizar los descriptores. Esto tuvo dos efectos directos, en primer lugar los tres algoritmos de clasificación disminuyeron su performance clasificando en promedio un 37 % de los patrones correctamente. Si bien el resultado es inferor al obtenido previamente, en esta ocación el algoritmo trabajó y desarrollo sus clasificadores como se ve en los resultados de la matriz de confusión.

Una vez con la base trabajada y los calsificadores trabajando como es esperable se optó por aplicar los algorítmos básicos de extracción de características, PCA y LDA. En el caso de PCA se optó por trabjar en el subespacio de los primeros cuatro componentes cuaya varianza acumulada supera al 80 %. Este cambio se vio reflejado en un aumento promedio del 5 % en la clasificación, pero más allá de la leve mejora esto nos lleva a pensar que no existe ningun tipo de distribución asociada a las clases. En la figura 3.1 se puede ver como los patrones presentan una distribución casi aleatoria respecto a las clases. Al aplicar LDA se redujo el espacio a una dimensión, la cual no presenta un poder de discriminación suficientemente superior a la observada anteriormente. Los resultados obtenidos son ligeramente superiores a los obtenidos mediante PCA salvo para el clasificador k-NN.

Finalmente los resultados demostraron que no es posible expresar, mediante los algorítmos aplicados, una correlación entre el gen estudiado y las variables fenotípicas. Naturalmente no podemos afirmar lo contrario, queda todavía un largo conjunto de herramientas estdísticas por aplicar, algunas de ellas con un enfoque mayor a las dadas en el curso hacia problemas biológicos.

También se comprobó como, que en general, cuando uno no puede visualizar la distribución de los datos o incluso las correlaciónes entre las variables medidas de modo visual, previo a cualquier tipo de análisis. Es muy dificil que un algoritmo tenga un buen desempeño, los algoitmos vistos en el curso buscan automatizar ese tipo de análisis que uno puede realizar y hasta sospechar simplemente analizando los datos.

Bibliografía

[1] Eileen Armstrong Reborati, *Detección y análisis de genes asociados a la calidad de la carne en bovinos*, Tesis de doctorado, Madrid, 2011.

[2] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11.

[3] Meyer, K. (2007). WOMBAT – A tool for mixed model analyses in quantitative genetics by REML, J. Zhejiang Uni.

[4] Ronald R. Coifman y Stéphane Lafon, *Applied and Computational Harmonic Analysis*, Volume 21, Issue 1, July 2006, Pages 5–30.