Bayes ingenuo

Por:
Ian Gabriel Cañas Fernández, 1092228

.

Profesor: Juan S. Pérez R.,
Asignatura: INL367L, Secc 01

Resumen:

En este laboratorio se estará implementando los conocimientos del clasificador de Naive Bayes aplicado a variables discretas, en las que se reconocerá la frecuencia de estos para así poder predecir si un pasajero a bordo al momento de la catástrofe del RMS Titanic sobrevivió o no al evento, para ello se considerará los datos obtenidos para cada pasajero, aplicando el teorema de Bayes para el acierto del clasificador.

Ejercicios previos:

P5. Predecir la supervivencia.

En un dataset dado se solicita clasificar un set de datos de tripulantes del Titanic para reconocer si estos han sobrevivido o no, para ello, lo primero solicitado es su predicción mediante el uso de un clasificador Naive Bayes reconociendo que varias variables son discretas. La selección de las características asumió lo siguiente:

- EL ID del pasajero, su nombre, cabina y ticket son características únicas por pasajero.
- Pclass y Sex, son variables discretas representando la clase del boleto y el sexo de la persona, pues eran características representativas a la hora de seleccionar la supervivencia de las personas en contextos catastróficos. Incluyendo a la hora de seleccionar prioridades.
- La edad de la persona se ignorará, pues, aunque pudo haber prioridad por edades, no representa una distribución lo suficientemente simple al modelado.
- La frecuencia de los valores Embarked será ignorada porque la catástrofe aconteció luego de las embarcaciones a bordo, caso contrario no se presentaren en el listado.
- Se selecciona la cantidad de hermano y familiares, SibSp y Parch.

E5.1 Naive Bayes; predicción de la supervivencia.

El primer método de machine learning utilizado para predecir la supervivencia de un grupo de personas se ha hecho una consideración de la frecuencia de personas que cumplen ciertas características y hayan resultado tanto muertas como vivas. Para no hacer un conteo para todas las posibles combinaciones, hacemos uso del teorema de bayes para más de una característica mediante la siguiente expresión:

$$P(C_K|x_1, x_2, \dots, x_d) = P(C_K|\boldsymbol{x})$$
$$= \frac{P(C_K)P(\boldsymbol{x}|C_K)}{P(\boldsymbol{x})}$$

Para el uso de las frecuencias se trabajó con los histogramas de cada característica según la clase que, en el caso, es la supervivencia (o no) de la persona. Calculamos las probabilidades a priori:

```
PVivo = sum(traindata.Survived==1)/length(traindata.Survived);
PMuerto = sum(traindata.Survived==0)/length(traindata.Survived);
```

Los histogramas para cada clase individual

```
%% Survived, Pclass, SibSp, Parch.
% sex,

liveMen = sum(vivos.SexN==0)/length(vivos.SexN);
liveWomen = sum(vivos.SexN==1)/length(vivos.SexN);

deathMen = sum(muertos.SexN==0)/length(muertos.SexN);
deathWomen = sum(muertos.SexN==1)/length(muertos.SexN);

[vivo_freqSex vivo_NSex] = hist(vivos.SexN, min(vivos.SexN):1:max(vivos.SexN));
vivo_freqSex = vivo_freqSex./length(vivos.SexN);

[muerto_freqSex muerto_NSex] = hist(muertos.SexN, min(muertos.SexN):1:max(muertos.SexN));
muerto_freqSex = muerto_freqSex./length(muertos.SexN);


%% Histogramas para gente viva

[vivo_freqPclass vivo_NPclass] = hist(vivos.Pclass, min(vivos.Pclass):1:max(vivos.Pclass));
[vivo_freqSibSp vivo_NSibSp] = hist(vivos.SibSp, min(vivos.SibSp):1:max(vivos.SibSp)*2);
[vivo_freqParch vivo_NParch] = hist(vivos.Parch, min(vivos.Parch):1:max(vivos.Parch)*3);

vivo_freqPclass = vivo_freqPclass./length(vivos.Pclass);
vivo_freqSibSp = vivo_freqSibSp./length(vivos.SibSp);
vivo_freqParch = vivo_freqParch./length(vivos.Parch);

%% Histogramas para gente viva

[muerto_freqPclass muerto_NPclass] = hist(muertos.Pclass, min(muertos.Pclass):1:max(muertos.Pclass));
[muerto_freqSibSp muerto_NSibSp] = hist(muertos.SibSp, min(muertos.SibSp):1:max(muertos.SibSp));
[muerto_freqParch muerto_NParch] = hist(muertos.Parch, min(muertos.Parch):1:max(muertos.Parch)*3);

muerto_freqPclass = muerto_freqPclass./length(muertos.Pclass);
muerto_freqSibSp = muerto_freqSibSp./length(muertos.SibSp);
muerto_freqParch = muerto_freqParch./length(muertos.Parch);
```

Finalmente, conociendo las frecuencias, encontramos los likelihood de supervivencia para cada una de las características:

```
%% Likelihood por sexo
minitest = testdata(:,:);
ulu = mod(find(vivos.SexN(:)==minitest.SexN'),length(traindata.PassengerId))+1;
% ulu = reshape(ulu, length(traindata.PassengerId), [])
```

```
i = vivo_NSex(:)==minitest.SexN';
i = mod(find(i)-1, length(vivo_NSex))+1;
likeVivoSex = vivo_freqSex(i);

i = muerto_NSex(:)==minitest.SexN';
i = mod(find(i)-1, length(muerto_NSex))+1;
likeMuertoSex = muerto_freqSex(i);
```

```
%% Likelihood por Pclass

i = vivo_NPclass(:)==minitest.Pclass';
i = mod(find(i)-1, length(vivo_NPclass))+1;
likeVivoPclass = vivo_freqPclass(i);

i = muerto_NPclass(:)==minitest.Pclass';
i = mod(find(i)-1, length(muerto_NPclass))+1;
likeMuertoPclass = muerto_freqPclass(i);
```

```
%% Likelihood por SibSp

i = vivo_NSibSp(:)==minitest.SibSp';
i = mod(find(i)-1, length(vivo_NSibSp))+1;
likeVivoSibSp = vivo_freqSibSp(i);

i = muerto_NSibSp(:)==minitest.SibSp';
i = mod(find(i)-1, length(muerto_NSibSp))+1;
likeMuertoSibSp = muerto_freqSibSp(i);
```

```
%% Likelihood por Parch

i = vivo_NParch(:)==minitest.Parch';
i = mod(find(i)-1, length(vivo_NParch))+1;
likeVivoParch = vivo_freqParch(i);

i = muerto_NParch(:)==minitest.Parch';
i = mod(find(i)-1, length(muerto_NParch))+1;
likeMuertoParch = muerto_freqParch(i);
```

Luego, aplicamos el teorema de bayes y comprobamos su eficiencia:

```
gVivo = PVivo.*likeVivoSex.*likeVivoPclass.*likeVivoSibSp.*likeVivoParch;
gMuerto = PMuerto.*likeMuertoSex.*likeMuertoPclass.*likeMuertoSibSp.*likeMuertoParch;

MLSurvived = nan(length(minitest.PassengerId),1);
minitest = [minitest table(MLSurvived)];

minitest.MLSurvived(:) = gVivo > gMuerto;
NaiveAccuracy = sum(minitest.MLSurvived==tester.Survived(:))/length(tester.Survived)

ind = tester.Survived(find(minitest.MLSurvived~=tester.Survived(:)));
falseNegative = sum(ind)/length(tester.Survived)
falsePositive = sum(~ind)/length(tester.Survived)
```

```
%%
% Si observamos los casos en que no se ha acertado el valor de
% supervivencia podemos observar que se presentan outliers, por lo que
% puede ser considerada una razón por la que tenemos fallos en los
% aciertos.

minitest(find(minitest.MLSurvived~=tester.Survived(:)), :)
```

E5.2 Implementación en Matlab de kNN

De otra manera, se ha procurado hacer la predicción de supervivencia en base a las características, considerando la distancia euclideana de hacia las distintas clases. A continuación, se presenta el código en Matlab del algoritmo conocido implementado en Matlab.

```
%%

MLSurvived = nan(length(testdata.PassengerId),1);
testdata = [testdata table(MLSurvived)];

for k = 1:2:7
    for i = 1:length(testdata.PassengerId)
        i;
        d = ([testdata.SexN(i), testdata.Pclass(i), testdata.SibSp(i), testdata.Parch(i)] - [traindata.SexN(:), traindata.Pclass(:), traindata.SibSp(:), traindata.Parch(:)]).^2;
        d = sqrt(d(:,1)+d(:,2)+d(:,3)+d(:,4));

        [dis, pos] = mink(d, k);
        testdata.MLSurvived(i) = mode(traindata.Survived(pos));
    end
    k
    NaiveAccuracy = sum(testdata.MLSurvived==tester.Survived(:))/length(tester.Survived)
end
```

R5.1 Resultados:

Para confirmar el rendimiento del algoritmo utilizaremos la métrica de tasa de aciertos de la predicción respecto a los datos obtenidos más tarde. Se presentan los resultados vistos por Naive Bayes y los vistos para kNN con diferentes cantidades de (k) vecinos cercanos.

```
NaiveAccuracy =

    0.9522


falseNegative =

    0.0144


falsePositive =

    0.0335

Elapsed time is 0.750710 seconds.
```

Ilustración 1. Resultados de Naive Bayes.

```
k =

     1

kNNAccuracy =

    0.6914

k =

     3

kNNAccuracy =

    0.5239

k =

     5

kNNAccuracy =

    0.9139

k =

     7

kNNAccuracy =

    0.9378

Elapsed time is 0.282120 seconds.
```

Ilustración 2. Resultados para varias cantidades de vecinos mediante kNN.

A.5 Análisis:

En base a los resultados obtenidos mediante los dos métodos de clasificación, se pudo observar que el uso de Naive Bayes como clasificador es el más indicado en cuanto a características discretas se refiere, especialmente si estamos hablando de variables categóricas, pues este método es ciego al orden. Comparando los resultados se observa que, para mayores valores de k en kNN, mayor la eficiencia; sin embargo, en ningún caso kNN supera la eficiencia vista por Naive Bayes: un 95.22 %.