$\LaTeX$ command declarations here.

EECS 445: Machine Learning

Hands On 08: More on Naive Bayes (saving SVM for later...)

- Instructor: **Ben Bray, Chansoo Lee, Jia Deng, Jake Abernethy**
- Date: October 5, 2016

Estimating the probability of a lead pipes Flint

Officials are investing $27Million to dig up lead pipes in Flint MI. Before they spend $5k on digging up the pipes for a home, they want a better estimate whether the pipe is lead. We have two observable variables, whether the home is *Old* (i.e. built before 1950) or not (built after 1950), and what the messy city records suggest. Keep in mind the city records are often wrong.

We make the "naive bayes" assumption that, given the target HasLead(X), the events IsOld(X) and RecordsSayLead(X) are independent of each other. Initially, the city believes the following parameters are roughly true:

$$\begin{aligned} P(HasLead(X)) &= 0.4 \\ P(IsOld(X) \mid HasLead(X)) &= 0.7 \\ P(IsOld(X) \mid NotHasLead(X)) &= 0.3 \\ P(RecordsSayLead(X) \mid HasLead(X)) &= 0.8 \\ P(RecordsSayLead(X) \mid NotHasLead(X)) &= 0.5 \end{aligned}$$

Compute the probabilty:

$$P(HasLead(X) \mid IsOld(X), RecordsSayLead(X))$$

Now do the same for the other three conditions (i.e. conditioning on $IsOld(X) \& NotRecordsSayLead(X)$, etc.)

Solution:

We use Bayes Rule, then use the **Naive Bayes assumption**.

$$\begin{aligned} P(HasLead(X) \mid IsOld(X), RecordsSayLead(X)) &= \frac{P(IsOld(X), RecordsSayLead(X) \mid HasLead(X)) \cdot P(}{P(IsOld(X), RecordsSayLead(X))} \\ &= \frac{P(IsOld(X) \mid HasLead(X)) \cdot P(RecordsSayLead(X) \mid}{P(IsOld(X), RecordsSayLea} \\ &= \frac{0.7 \cdot 0.8 \cdot 0.4}{P(IsOld(X), RecordsSayLead(X))} \end{aligned}$$

We also have to marginalize to compute the denominator.

$$\begin{aligned} P(IsOld(X), RecordsSayLead(X)) &= P(IsOld(X), RecordsSayLead(X), HasLead(X)) \\ &\quad + P(IsOld(X), RecordsSayLead(X), NotHasLead(X)) \\ &= P(IsOld(X), RecordsSayLead(X) \mid HasLead(X)) \cdot P(HasLead(X)) \\ &\quad + P(IsOld(X), RecordsSayLead(X) \mid NotHasLead(X)) \cdot P(NotHa \\ &= P(IsOld(X) \mid HasLead(X)) \cdot P(RecordsSayLead(X) \mid HasLead(X)) \cdot \\ &\quad + P(IsOld(X) \mid NotHasLead(X)) \cdot P(RecordsSayLead(X) \mid NotH \\ &= 0.7 \cdot 0.8 \cdot 0.4 + 0.3 \cdot 0.5 \cdot (1 - 0.4) \end{aligned}$$

To the final answer is $\frac{0.7 \cdot 0.8 \cdot 0.4}{0.7 \cdot 0.8 \cdot 0.4 + 0.3 \cdot 0.5 \cdot (1-0.4)} \approx 0.713.$

Flint Starts Gathering Data, wants to update parameters

Over the past month, Flint has dug up about 200 service lines, and they've observed the pipe materials for several of these. They are starting to believe their initial estimates are incorrect.

They want to update these values, still assuming the Naive Bayes model. Here are the necessary parameters of this model: \begin{align} P(HasLead(X)) &= \pi*{\text{HasLead}} = ?\ P(IsOld(X) \mid HasLead(X)) & = \theta*{\text{HasLead}, \text{IsOld}} = ?\ P(IsOld(X) \mid Not HasLead(X)) &= \theta*{\text{NotHasLead}, \text{IsOld}} = ?\ P(RecordsSayLead(X) \mid HasLead(X)) &= \theta*{\text{HasLead}, \text{RecordsSayLead}} = ?\ P(RecordsSayLead(X) \mid Not HasLead(X)) &= \theta_{\text{NotHasLead}, \text{RecordsSayLead}} = ? \end{align}

Load the dataset and compute the **maximum likelihood** estimate for the above parameters.

In [1]:

```
%pylab inline
import numpy as np
import pandas as pd
data = pd.read_csv('estimating_lead.csv')

# Run this to see a printout of the data
data
```

```
Populating the interactive namespace from numpy and matplotlib
```

Out[1]:

	Lead	IsOld	RecordSaysLead
0	Lead	True	True
1	Lead	True	True
2	Lead	True	True
3	Lead	True	True
4	Lead	True	True
5	NotLead	True	True
6	Lead	True	True
7	NotLead	False	False
8	Lead	True	True
9	Lead	True	True
10	Lead	True	True
11	NotLead	False	False
12	NotLead	True	False
13	Lead	True	False
14	Lead	True	True
15	Lead	True	True
16	Lead	True	True
17	Lead	True	True
18	Lead	True	False
19	Lead	True	True
20	Lead	True	True
21	NotLead	False	True
22	Lead	True	False
23	Lead	True	False
24	Lead	True	False
25	NotLead	False	True
26	NotLead	False	False
27	Lead	True	True
28	Lead	True	True
29	Lead	False	True
...	...	...	...
170	NotLead	True	True

	Lead	IsOld	RecordSaysLead
171	NotLead	True	True
172	NotLead	False	False
173	Lead	True	True
174	Lead	True	True
175	Lead	False	True
176	NotLead	True	True
177	Lead	True	True
178	Lead	True	True
179	NotLead	False	False
180	Lead	True	True
181	Lead	False	True
182	NotLead	True	False
183	Lead	True	True
184	Lead	True	True
185	Lead	True	True
186	Lead	False	True
187	Lead	True	True
188	NotLead	False	False
189	Lead	True	True
190	NotLead	False	False
191	Lead	True	True
192	Lead	True	True
193	Lead	True	False
194	NotLead	True	False
195	Lead	True	True
196	Lead	False	True
197	Lead	False	True
198	Lead	False	True
199	Lead	True	True

200 rows × 3 columns

```
In [2]: # The object 'data' is a pandas DataFrame
        # Don't worry if you don't know what that is, we can turn it into a nump
        y array
        datamatrix = data.as_matrix()
```

Solution

For the Naive Bayes model, computing the MLE estimate for the parameters is pretty easy because it can be reduced to estimating empirical frequencies (i.e. you just need to compute the "count" of how many times something occured in our dataset). Let's say we have N examples in our dataset, and I'll use the notation $\#\{\}$ to mean "size of set".

$$\begin{aligned}
\pi^{\text{MLE}}_{\text{HasLead}} &= \frac{\#\{HasLead(X_i)\}}{N} \\
\theta^{\text{MLE}}_{\text{HasLead,IsOld}} &= \frac{\#\{HasLead(X_i) \wedge IsOld(X_i)\}}{\#\{HasLead(X_i)\}} \\
\theta^{\text{MLE}}_{\text{Not HasLead,IsOld}} &= \frac{\#\{NotHasLead(X_i) \wedge IsOld(X_i)\}}{\#\{NotHasLead(X_i)\}} \\
\theta^{\text{MLE}}_{\text{HasLead,RSL}} &= \frac{\#\{HasLead(X_i) \wedge RecordSaysLead(X_i)\}}{\#\{HasLead(X_i)\}} \\
\theta^{\text{MLE}}_{\text{Not HasLead,RSL}} &= \frac{\#\{NotHasLead(X_i) \wedge RecordSaysLead(X_i)\}}{\#\{NotHasLead(X_i)\}}
\end{aligned}$$

In [14]:
```
# We can use some pandas magic to do these counts quickly

N = data.shape[0]
params = {
    'pi_mle': data[data.Lead == 'Lead'].shape[0] / N,
    'theta_haslead_isold': data[(data.Lead == 'Lead') & (data.IsOld == T
rue) ].shape[0] / \
                                data[data.Lead == 'Lead'].shape[0],
    'theta_nothaslead_isold': data[(data.Lead != 'Lead') & (data.IsOld =
= True) ].shape[0] / \
                                data[data.Lead != 'Lead'].shape[0],
    'theta_haslead_rsl': data[(data.Lead == 'Lead') & (data.RecordSaysLe
ad == True) ].shape[0] / \
                                data[data.Lead == 'Lead'].shape[0],
    'theta_nothaslead_rsl': data[(data.Lead != 'Lead') & (data.RecordSay
sLead == True) ].shape[0] / \
                                data[data.Lead != 'Lead'].shape[0],
}
print(pd.Series(params))
```

```
pi_mle                    0.700000
theta_haslead_isold       0.778571
theta_haslead_rsl         0.900000
theta_nothaslead_isold    0.383333
theta_nothaslead_rsl      0.433333
dtype: float64
```

Putting a Prior on π_{HasLead}

For the case of the discreet event, such as material=Lead or =NoLead, we are working with a categorical distribution, i.e. a distrbution on one of C things occuring. The parameters of this distribution are a probability vector $\pi \in \Delta_C$. (That is, $\pi_c \geq 0$ for all c, and $\sum_c \pi_c = 1$.)

Often when we have limited data, we want to add a prior distribution on our parameters. The standard prior to use is a *Dirichlet* with parameters $\alpha_1, \ldots, \alpha_C$. That is, we assume that $\pi \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_C)$. Recall that the Dirichlet has PDF $f(\pi_1, \ldots, \pi_C) = \frac{1}{B(\alpha)} \prod_{c=1}^{C} \pi_c^{\alpha_c - 1}$, where $B(\cdot)$ is just the normalizing term.

For our Flint problem, assume that the parameters $(\pi_{\text{HasLead}}, 1 - \pi_{\text{HasLead}}) \sim \text{Dirichlet}(3, 3)$. Compute the MAP estimate of π_{HasLead} for this distribution using the above dataset.

Solution

For the Naive Bayes model, computing the MAP estimate is very similar to the MLE, but you have to add the Dirichlet prior parameters as "pseudocounts" to the frequency calculation. In this case we have two prior parameters α_{HasLead} and $\alpha_{\text{NotHasLead}}$, which we are choosing to be the value 3. \begin{align} \pi_{\text{HasLead}}^{\text{MAP}} &= \frac{#{HasLead(X*i)}* + *alpha*{\text{HasLead}} - 1 }{N + \alpha*{\text{HasLead}} - 1* + *alpha*{\text{NotHasLead}} - 1} = \frac{#{HasLead(X_i)} + 2 }{N + 4} \end{align}