customer_segments

March 24, 2016

1 Creating Customer Segments

In this project you, will analyze a dataset containing annual spending amounts for internal structure, to understand the variation in the different types of customers that a wholesale distributor interacts with.

Instructions:

- Run each code block below by pressing **Shift+Enter**, making sure to implement any steps marked with a TODO.
- Answer each question in the space provided by editing the blocks labeled "Answer:".
- When you are done, submit the completed notebook (.ipynb) with all code blocks executed, as well as a .pdf version (File > Download as).

```
In [2]: # Import libraries: NumPy, pandas, matplotlib
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt

        # Tell iPython to include plots inline in the notebook
        %matplotlib inline

        # Read dataset
        data = pd.read_csv("wholesale-customers.csv")
        print "Dataset has {} rows, {} columns".format(*data.shape)
        print data.head()  # print the first 5 rows
```

```
Dataset has 440 rows, 6 columns
   Fresh  Milk  Grocery  Frozen  Detergents_Paper  Delicatessen
0  12669  9656     7561     214              2674          1338
1   7057  9810     9568    1762              3293          1776
2   6353  8808     7684    2405              3516          7844
3  13265  1196     4221    6404               507          1788
4  22615  5410     7198    3915              1777          5185
```

1.1 Feature Transformation

1) In this section you will be using PCA and ICA to start to understand the structure of the data. Before doing any computations, what do you think will show up in your computations? List one or two ideas for what might show up as the first PCA dimensions, or what type of vectors will show up as ICA dimensions.

Answer: The first dimension of PCA will compose a new feature by using the old features and maximize the variance dimension, so according to the variances of each original feature values, the first dimension is likely to be composed by "Fresh" and "Grocery". In ICA dimesions, any two vectors vertical, $v_i \wedge v_j = 0$, so we can have independent features. After transformation if in one dimension the variance is small

1.1.1 PCA

In [118]:
```
# Apply PCA with the same number of dimensions as variables in the dataset
from sklearn.decomposition import PCA
pca = PCA(n_components=6)
pca.fit(data)

# Print the components and the amount of variance in the data contained in each dimension
print data.columns
print pca.components_
print pca.explained_variance_ratio_
print data.var()
```

```
Index([u'Fresh', u'Milk', u'Grocery', u'Frozen', u'Detergents_Paper',
       u'Delicatessen'],
      dtype='object')
[[-0.97653685 -0.12118407 -0.06154039 -0.15236462  0.00705417 -0.06810471]
 [-0.11061386  0.51580216  0.76460638 -0.01872345  0.36535076  0.05707921]
 [-0.17855726  0.50988675 -0.27578088  0.71420037 -0.20440987  0.28321747]
 [-0.04187648 -0.64564047  0.37546049  0.64629232  0.14938013 -0.02039579]
 [ 0.015986    0.20323566 -0.1602915   0.22018612  0.20793016 -0.91707659]
 [-0.01576316  0.03349187  0.41093894 -0.01328898 -0.87128428 -0.26541687]]
[ 0.45961362  0.40517227  0.07003008  0.04402344  0.01502212  0.00613848]
Fresh               1.599549e+08
Milk                5.446997e+07
Grocery             9.031010e+07
Frozen              2.356785e+07
Detergents_Paper    2.273244e+07
Delicatessen        7.952997e+06
dtype: float64
```

2) How quickly does the variance drop off by dimension? If you were to use PCA on this dataset, how many dimensions would you choose for your analysis? Why?

Answer: It begins to drop a lot at the third dimension. I will choos the first two since it will cover about 0.86 of the variance.

3) What do the dimensions seem to represent? How can you use this information?

Answer: In original data we have six features: Fresh, Milk, Grocery, Frozen, Detergents_Paper, Delicatessen, but we can compose them by using PCA and use new components to represent the data. The first feature value in new dataset is determined by the first row of the components matrix. So we can see the first new feature is mainly composed by Fresh. The second is mainly composed by Milk, Grocery and Detergent paper.

If we multiply our data set with only the first two rows of the components matrix, we can have a new dataset with only two principle features and kind of maintain the information.

1.1.2 ICA

In [63]:
```
# Fit an ICA model to the data
# Note: Adjust the data to have center at the origin first!
from sklearn.decomposition import FastICA

data_center = data - data.mean()

ica = FastICA(n_components=6)
ica.fit(data_center)
# Print the independent components
print ica.components_
```

```
[[ -3.97584552e-06   8.58485649e-07   6.21134235e-07   6.77711688e-07
   -2.05433190e-06   1.04510932e-06]
 [  8.65235656e-07   1.40333824e-07  -7.74233463e-07  -1.11461250e-05
    5.55718437e-07   5.95237484e-06]
 [  1.53507603e-07   9.84597768e-06  -5.80801858e-06  -3.64196027e-07
    3.30984386e-06  -6.05839856e-06]
 [ -3.00418561e-07   2.30078823e-06   1.20765692e-05  -1.46181304e-06
   -2.82096973e-05  -5.73309406e-06]
 [  3.86407386e-07   2.19528255e-07   6.01170445e-07   5.22078127e-07
   -5.10531816e-07  -1.80927093e-05]
 [ -2.10926597e-07   1.89056510e-06  -6.39606016e-06  -4.15093483e-07
    7.37227705e-07   1.43958933e-06]]
```

4) For each vector in the ICA decomposition, write a sentence or two explaining what sort of object or property it corresponds to. What could these components be used for?

Answer: this matrix is the unmixing matrix. By multiply the data matrix from the original data with the unmixing matrix, we can get a new dataset and each new feature is independent with each other.

1.2 Clustering

In this section you will choose either K Means clustering or Gaussian Mixed Models clustering, which implements expectation-maximization. Then you will sample elements from the clusters to understand their significance.

1.2.1 Choose a Cluster Type

5) What are the advantages of using K Means clustering or Gaussian Mixture Models?

Answer: K Means advantage: - if data set is huge, then K-Means most of the times computationally faster. It doesn't need to compute the expectation of each data belongs to each cluster in every iteration, instead only assign each data to one cluster in each iteration. - could be used as a pre-process method

Disadvatage: - NP hard, a lot of local maximum - final results relies on the initial cluster(s) - Hard to determine K

Gaussian Mixture models advatage: - Gaussian Mixture Models can be be seen as a generalization of k-means. It can express the uncertainty of one instance belongs to different clusters.

Disadvatage: - more expensive to compute then K Means because of using EM

6) Below is some starter code to help you visualize some cluster data. The visualization is based on this demo from the sklearn documentation.

```
In [64]: # Import clustering modules
         from sklearn.cluster import KMeans
         from sklearn.mixture import GMM
```

```
In [107]: # First we reduce the data to two dimensions using PCA to capture variation
          reduced_data = PCA(n_components=2).fit_transform(data)
          print reduced_data[:10]  # print upto 10 elements
```

```
[[  -650.02212207   1585.51909007]
 [  4426.80497937   4042.45150884]
 [  4841.9987068    2578.762176  ]
 [  -990.34643689  -6279.80599663]
 [-10657.99873116  -2159.72581518]
 [  2765.96159271   -959.87072713]
 [   715.55089221  -2013.00226567]
 [  4474.58366697   1429.49697204]
 [  6712.09539718  -2205.90915598]
 [  4823.63435407  13480.55920489]]
```

In [108]:
```
# TODO: Implement your clustering algorithm here, and fit it to the reduced data for visualiz
# The visualizer below assumes your clustering object is named 'clusters'
from sklearn import mixture
from sklearn.cluster import KMeans
clusters = mixture.GMM(n_components=2)
#clusters = KMeans(n_clusters=2)
clusters.fit(reduced_data)
print clusters
```

```
GMM(covariance_type='diag', init_params='wmc', min_covar=0.001,
  n_components=2, n_init=1, n_iter=100, params='wmc', random_state=None,
  thresh=None, tol=0.001, verbose=0)
```

In [109]:
```
# Plot the decision boundary by building a mesh grid to populate a graph.
x_min, x_max = reduced_data[:, 0].min() - 1, reduced_data[:, 0].max() + 1
y_min, y_max = reduced_data[:, 1].min() - 1, reduced_data[:, 1].max() + 1
hx = (x_max-x_min)/1000.
hy = (y_max-y_min)/1000.
xx, yy = np.meshgrid(np.arange(x_min, x_max, hx), np.arange(y_min, y_max, hy))

# Obtain labels for each point in mesh. Use last trained model.
Z = clusters.predict(np.c_[xx.ravel(), yy.ravel()])
```

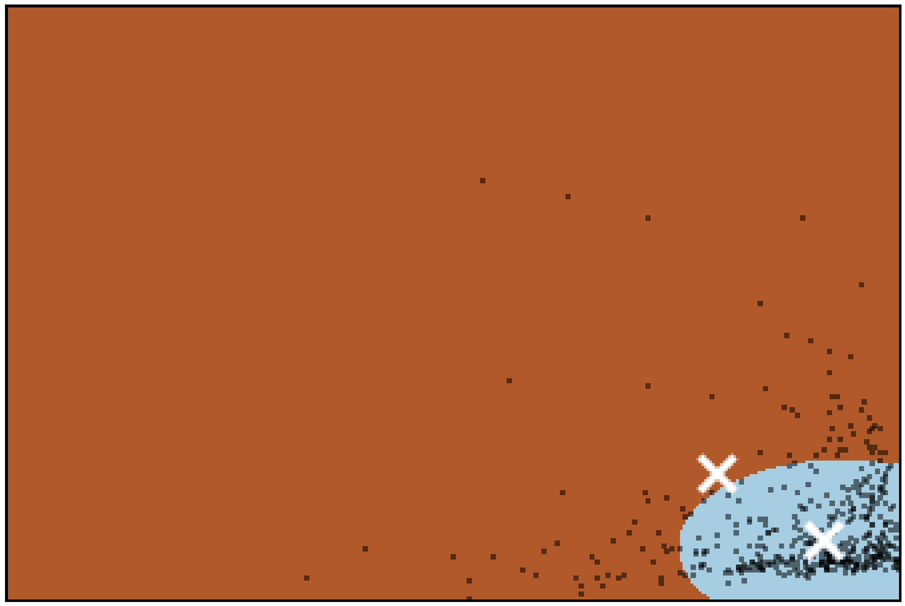
In [110]:
```
# TODO: Find the centroids for KMeans or the cluster means for GMM

centroids = clusters.means_
#centroids = clusters.cluster_centers_
print centroids
```

```
[[  3308.39301792  -3017.01739698]
 [-10810.23008886   9858.15532401]]
```

In [111]:
```
# Put the result into a color plot
Z = Z.reshape(xx.shape)
plt.figure(1)
plt.clf()
plt.imshow(Z, interpolation='nearest',
           extent=(xx.min(), xx.max(), yy.min(), yy.max()),
           cmap=plt.cm.Paired,
           aspect='auto', origin='lower')

plt.plot(reduced_data[:, 0], reduced_data[:, 1], 'k.', markersize=2)
plt.scatter(centroids[:, 0], centroids[:, 1],
            marker='x', s=169, linewidths=3,
            color='w', zorder=10)
plt.title('Clustering on the wholesale grocery dataset (PCA-reduced data)\n'
          'Centroids are marked with white cross')
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks(())
plt.yticks(())
plt.show()
```



7) What are the central objects in each cluster? Describe them as customers.

Answer: Central objects are near the centroids, in the figure above, the lower centraid represents a kind of customers who spends less. The upper one represents those customers who spend a lot. The data also shows that customers on the upper cluster tend to spend more on one dimention then the other.

1.2.2 Conclusions

** 8)** Which of these techniques did you feel gave you the most insight into the data?

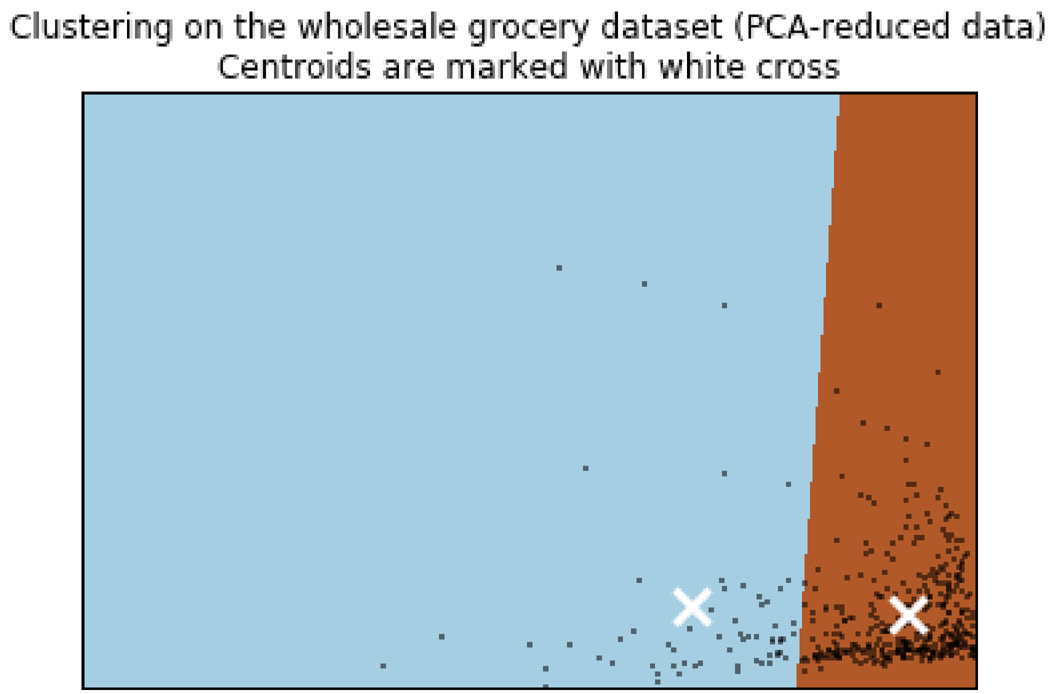
In [112]:
```
from sklearn import mixture
from sklearn.cluster import KMeans
clusters = KMeans(n_clusters=2)
clusters.fit(reduced_data)
print clusters
x_min, x_max = reduced_data[:, 0].min() - 1, reduced_data[:, 0].max() + 1
y_min, y_max = reduced_data[:, 1].min() - 1, reduced_data[:, 1].max() + 1
hx = (x_max-x_min)/1000.
hy = (y_max-y_min)/1000.
xx, yy = np.meshgrid(np.arange(x_min, x_max, hx), np.arange(y_min, y_max, hy))

# Obtain labels for each point in mesh. Use last trained model.
Z = clusters.predict(np.c_[xx.ravel(), yy.ravel()])
centroids = clusters.cluster_centers_
# Put the result into a color plot
Z = Z.reshape(xx.shape)
plt.figure(1)
plt.clf()
plt.imshow(Z, interpolation='nearest',
           extent=(xx.min(), xx.max(), yy.min(), yy.max()),
```

```
                 cmap=plt.cm.Paired,
                 aspect='auto', origin='lower')

    plt.plot(reduced_data[:, 0], reduced_data[:, 1], 'k.', markersize=2)
    plt.scatter(centroids[:, 0], centroids[:, 1],
                marker='x', s=169, linewidths=3,
                color='w', zorder=10)
    plt.title('Clustering on the wholesale grocery dataset (PCA-reduced data)\n'
              'Centroids are marked with white cross')
    plt.xlim(x_min, x_max)
    plt.ylim(y_min, y_max)
    plt.xticks(())
    plt.yticks(())
    plt.show()
```

```
KMeans(copy_x=True, init='k-means++', max_iter=300, n_clusters=2, n_init=10,
    n_jobs=1, precompute_distances='auto', random_state=None, tol=0.0001,
    verbose=0)
```



Answer: I feel PCA plays a huge role here to reduce the feature dimensions to 2 and groups the dataset together in order to cluster. After using PCA, we can apply KMeans or GMM. And GMM seems to be better here since there's no explicit line here for seperate two clusters.

9) How would you use that technique to help the company design new experiments?

Answer: Since we can split our customers into two clusters, we may want to know which cluster gives the company larger revenue. So that we can pay more attention to what they need. We may design personalized coupons or promotions to customers from different clusters to see whether we can boost the sales.

10) How would you use that data to help you predict future customer needs?

Answer: The company may need to track more behaviors of customers from each cluster. For example small buyers may from the families and we find them tend to come during weekend, we may have some family promotions during that time. For large buyers we can estabish a department to keep track their needs and communicate with them frequently to adjust our purchase.