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 [113]:

```
# 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
```

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: If we use PCA(n_componets = n), we will obtain n vectors. These vectors have two important properties. First these n vectors are orthogonal each other. It means these vectors are part of "basis of feature data's dimension". The data set's dimension can represent with some basis. These vectors are the 'n' most important basis among the basis.

ICA will return independent components set. Basically they will maximize the statistical independence of the estimated components.

PCA

In [114]:

```python
# TODO: Apply PCA with the same number of dimensions as variables in the data set
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 pca.components_
print pca.explained_variance_ratio_
```

```
[[-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]
```

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: The value drops really quickly as components go backward. It depends on cost. I definitely choose first two major components, because it explains 85percent of variables. However, 86percent still seems too law to predict. So I believe that choosing three or four components is more proper which can explan 93% and 98%. If the time cost still has to spare, I can use all of them.

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

Answer: Let's talk about two major basis vectors. First one is [-0.97653685 -0.12118407 -0.06154039 -0.15236462 0.00705417 -0.06810471]. As we can see this one is almost similar to firstfeature(Fresh). So this information says that fresh_feature will be a important factor. Second major component is [-0.11061386 0.51580216 0.76460638 -0.01872345 0.36535076 0.05707921]. From this vector, we can check that the weighted sum of milk, grocery and detegernt_paper would be a important factor. We can rearrange or interpret data based on above criteria(component).

ICA

In [115]:

```
# TODO: Fit an ICA model to the data
# Note: Adjust the data to have center at the origin first!
from sklearn.decomposition import FastICA
ica = FastICA(n_components=6)
ica.fit(data-np.mean(data))

# Print the independent components
print ica.components_
```

```
[[  3.86439157e-07   2.19535245e-07   6.00759893e-07   5.22080411e-07
   -5.10176917e-07  -1.80925655e-05]
 [ -1.53637333e-07  -9.84533069e-06   5.80993042e-06   3.63852625e-07
   -3.31557259e-06   6.05745741e-06]
 [ -3.97591379e-06   8.59121773e-07   6.24633050e-07   6.77412582e-07
   -2.06186058e-06   1.04321208e-06]
 [  2.10555228e-07  -1.88679618e-06   6.42099113e-06   4.11898535e-07
   -7.94549778e-07  -1.45142846e-06]
 [ -2.99768771e-07   2.30639896e-06   1.20622589e-05  -1.46265385e-06
   -2.82069374e-05  -5.73201807e-06]
 [ -8.65200557e-07  -1.40445178e-07   7.74115697e-07   1.11461625e-05
   -5.55104455e-07  -5.95226352e-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: As I mentioned they are independent set, and they maximize the statistical independence. So It can be interpreted new feature of data set.

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.

Choose a Cluster Type

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

Answer: First let's talk about K means algorithm.

- (Pros) Simple and fast for data with low dimensionality.
- (Cons) K means cannot discern outliers.

Follwing properties are Gaussian EM models'.

- (Pros) Soft clustering is enabled. (calculate probablity of belonging to each group)
- (Pros) Obtain a density estimation for each cluster (also can discern outliers)
- (Cons) With a large data set, calculation can be slow.

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 [116]:

```
# Import clustering modules
from sklearn.cluster import KMeans
from sklearn.mixture import GMM
```

In [117]:

```
# TODO: First we reduce the data to two dimensions using PCA to capture varia
tion
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
pca.fit(data)

reduced_data = pca.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 [118]:

```
# TODO: Implement your clustering algorithm here, and fit it to the reduced d
ata for visualization
# The visualizer below assumes your clustering object is named 'clusters'
clusters = GMM(n_components=8)
clusters.fit(reduced_data)
print clusters
```

```
GMM(covariance_type='diag', init_params='wmc', min_covar=0.001,
  n_components=8, n_init=1, n_iter=100, params='wmc', random_state=None,
  thresh=None, tol=0.001, verbose=0)
```

In [119]:

```
# 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()])
```

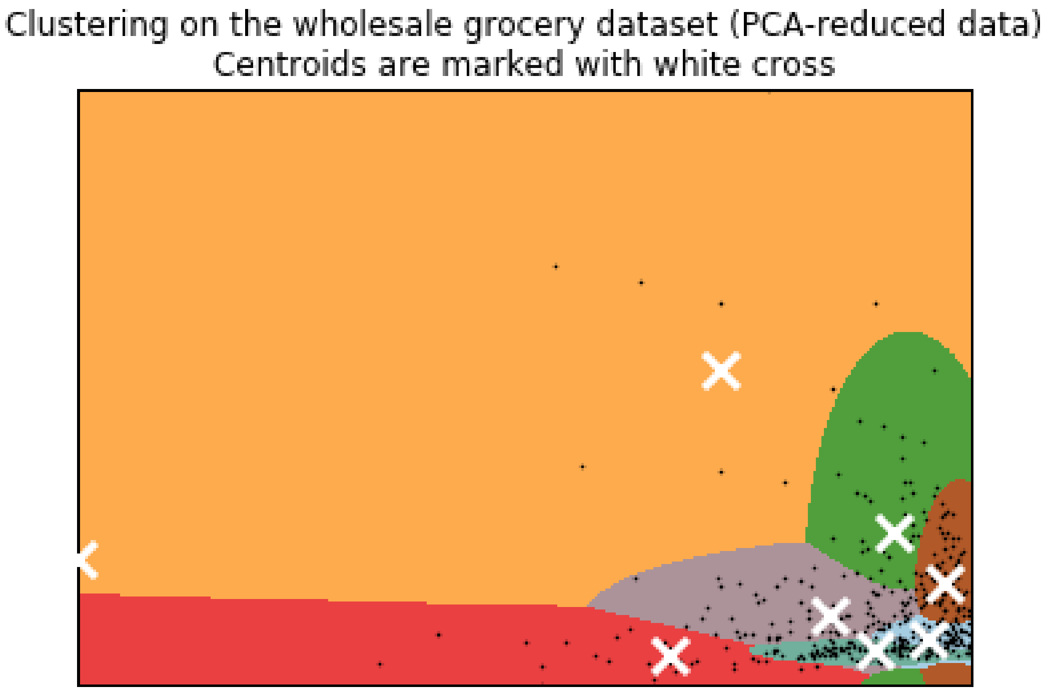
In [120]:

```
# TODO: Find the centroids for KMeans or the cluster means for GMM

centroids = clusters.means_
print centroids
```

```
[[   7536.85713524   -5271.65549509]
 [    454.96460816   -7661.51793015]
 [   2771.61549      14964.8735727 ]
 [ -26360.62555602   -8413.17639181]
 [ -19691.97729909   45688.76080332]
 [-103863.42532004    9910.34962857]
 [  -5560.93264448   -1033.10348831]
 [   9403.49190821    5422.96562892]]
```

In [121]:

```
# 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()
```



Clustering on the wholesale grocery dataset (PCA-reduced data)
Centroids are marked with white cross

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

Answer: These central points represent each clusters. I divided to 8 groups, but the most above one and most left group can be ignored (because they are quite outliers). So I would like to sperate to above 6 groups. I can say that two people who are belonging to same group have similar pattern of consumption. And also they will respond very similarly to some events.

Conclusions

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

Answer: I tried to use KMeans algorithm(even though I didn't include in this report). But the segmentation seems not that meaningful. I think the above segmentation seems very meaningful. Gaussian Mixture Model. It considers not only distance but also probability. So the GMM is more natural separation. On the other hands, KMeans algorithm's separation looks not meaningful. It just divide the area geometricaly.

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

Answer: Customer segmentation is really important task to marketing part. Using this data we can apply suitable advertisement or promotion to each segmentation group. For example, for green group customers, we can give them to sale promotion code or coupon of fresh food. Or delivery service for another group.

Also we can manage group respectively. If we manage whole people at once, there are many problems. Group with large number(like light green, brown group)'s tendency is much more powerfull than small groups tendency. So, in this case people of group(light green)'s dissatisfaction can be ignored by noise of major group. However using a segmentation and managing respectively can prevent these kinds of problems.

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

Answer: I will store data separately based on their group. I can take a suvey from people who belongs to different groups. And their opinion can represent their group. So I can predict their needs likewise.

In [ ]:

```
```