customer_segments

February 8, 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:

For PCA dimensions, my guess is that a combined component of "Fresh", "Milk", and "Grocery" will be the first principal component. In my opinion, "Fresh", "Milk", and "Grocery" all fall under the more general category of "groceries".

For ICA dimensions, my guess is that "Detergents_Paper" will be a basis vector of ICA. Out of all product categories (features) in our data, "Detergents_Paper" seems like the most different/independent category compared to the rest.

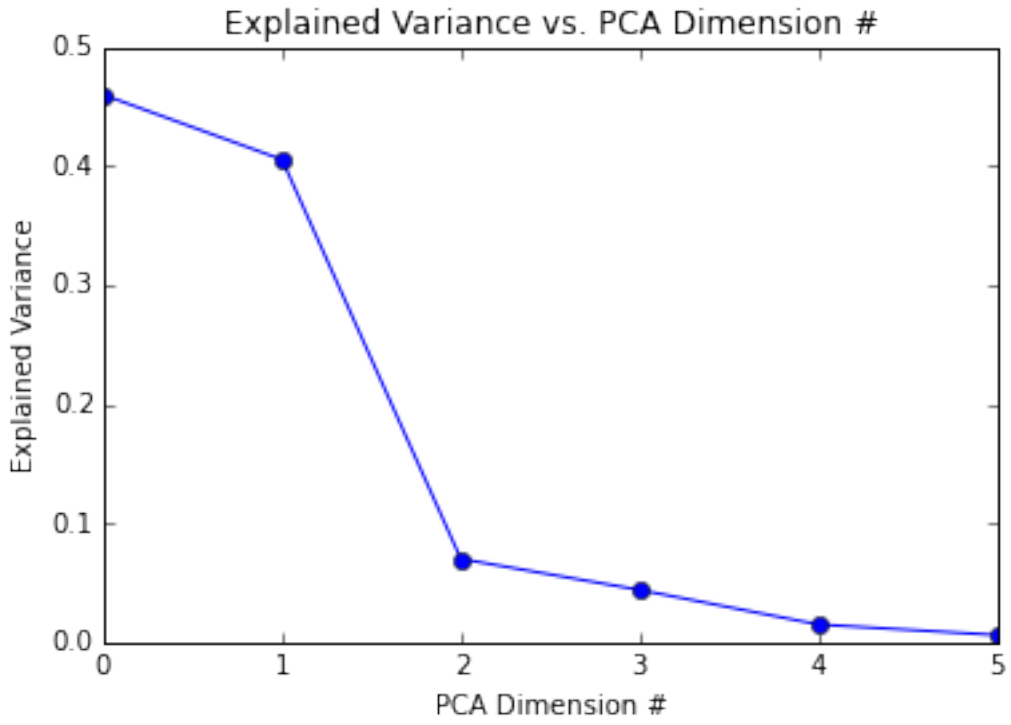
1.1.1 PCA

In [3]:
```
# TODO: Apply PCA with the same number of dimensions as variables in the dataset
from sklearn.decomposition import PCA

pca = PCA(n_components=len(data.columns))
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]
```

In [4]:
```
# Plot the explained variance as function of PCA dimension number
plt.plot(pca.explained_variance_ratio_, 'bo-')
plt.xlabel('PCA Dimension #')
plt.ylabel('Explained Variance')
plt.title('Explained Variance vs. PCA Dimension #')
```

Out[4]: <matplotlib.text.Text at 0x10abea690>



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:

Looking at PCA's explained variance ratio, we see that the variance drops off significantly after the second principle component/dimension. In particular, the first to second principle components explain 46% and 41% of the total variance, respectively. However, the third principle component only explains 7% of the total variance. From the above plot, we can clearly see the sharp drop-off in explained variance after the second principle component.

Based on the above reasoning, if I were to use PCA on this dataset, I would choose 2 dimensions for my analysis. Using only 2 dimensions also has an advantage of easy visualization, since 2-dimensional plots tend to be more intuitive for people to understand.

One thing to note is that if we had a pre-determined explained-variance threshold we want to exceed, we may choose a different number of dimensions to use. For example, if we wanted the reduced feature set to explain at least 90% of the variance, we would choose to use the first 3 principle components. The first three principle components would explain roughly 94% of the total variance (46% + 41% + 7% = 94%).

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

Answer:

The first dimension (first principle component) is mainly the "Fresh" feature, but negative. Looking at the first principle component, we see the first value (the value corresponding to "Fresh") is -0.9765, whereas all other values are much closer to 0. The fact that this value is negative is acceptable, as this means we found the appropriate eigenvector, just pointing in the opposite direction.

The second dimension (second principle component) seems to represent a weighted combination of "Grocery", "Milk", and "Detergents_Paper", since the values corresponding to the aforementioned features in the second principle component seem to be the largest.

From the analysis above, it seems we can determine our customer segments using just these two dimensions. Using the two principle components decribed above, we can run unsupervised clustering algorithms to discover customer segments.

Note that the "Frozen" and "Delicatessen" features do not play a large role in our reduced feature set, because the volume of sales in those categories are too low to create meaningful customer segments. If there is a particular category of customers that focus on purchasing "Frozen" or "Delicatessen" products, their volume is too low for us to allocate resouces to target them, relative to other customer categories.

1.1.2 ICA

```
In [19]: # TODO: Fit an ICA model to the data
         # Note: Adjust the data to have center at the origin first!
         from sklearn.decomposition import FastICA

         # Adjust the data to have center at the origin first
         data_norm = data.apply(lambda x: (x - np.mean(x)))

         # Since ICA produces random ordering of components and
         # random polarity of values within components, let's
         # fix the ordering and polarity across 3 separate FastICA runs
         for i in range(3):
             ica = FastICA()
             ica.fit(data_norm)

             # Scale ICA components such that max value is 1 in each component
             comp = np.apply_along_axis(lambda x: x/np.max(abs(x)), 1, ica.components_)

             # ICA gives arbitrary polarity for components,
             # so standardize the polarity such that
             # the first column's value is always positive
             for row in comp:
                 if row[0] < 0:
```

```
            row *= -1

        print 'Fixed Polarity ICA Run ' + str(i)
        # Display sorted ICA components, sorted by first value in component
        print np.round(comp[np.argsort(comp[:,0])], decimals=2)
        print '\n'
```

```
Fixed Polarity ICA Run 0
[[ 0.01 -0.08 -0.42  0.05  1.    0.21]
 [ 0.02  0.51 -0.27  0.    0.15 -1.  ]
 [ 0.02 -0.54  0.34  0.05 -0.16 -1.  ]
 [ 0.03 -0.25  1.    0.04 -0.37 -0.25]
 [ 0.08  0.02 -0.07 -1.    0.05  0.55]
 [ 1.   -0.17 -0.21 -0.17  0.6  -0.31]]


Fixed Polarity ICA Run 1
[[ 0.01 -0.09 -0.42  0.05  1.    0.2 ]
 [ 0.02  1.   -0.6  -0.04  0.35 -0.61]
 [ 0.02  0.01  0.03  0.03 -0.03 -1.  ]
 [ 0.03 -0.26  1.    0.05 -0.29 -0.24]
 [ 0.08  0.01 -0.07 -1.    0.05  0.53]
 [ 1.   -0.22 -0.18 -0.17  0.56 -0.25]]


Fixed Polarity ICA Run 2
[[ 0.01 -0.08 -0.43  0.05  1.    0.2 ]
 [ 0.02  1.   -0.59 -0.04  0.34 -0.62]
 [ 0.02  0.01  0.03  0.03 -0.03 -1.  ]
 [ 0.03 -0.29  1.    0.06 -0.12 -0.23]
 [ 0.08  0.01 -0.07 -1.    0.05  0.53]
 [ 1.   -0.22 -0.16 -0.17  0.52 -0.26]]
```

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:

First, looking at results from "Fixed Polarity ICA Run [0-2]", we see that all 6 vectors in the unmixing matrix are roughly consistent across all 3 runs of FastICA. Since FastICA is a stochastic process, we do expect small variation in the results, and such variation is acceptable.

In each vector, values with higher magnitude means that the archetypal customer is strongly affected by the corresponding feature. Each vector has a maximum magnitude of 1 for all its values. For example, the vector [0.08 0.01 -0.07 -1. 0.05 0.53] implies the archetypal customer is affected most greatly by frozen products, and affected the least by milk products.

Looking at "Fixed Polarity ICA Run 2", let's examine each vector within the unmixing matrix.

The vector [0.01 -0.08 -0.43 0.05 1. 0.2] represents a customer segment that is mostly affected by detergents/paper products, followed by grocery products, and a bit of delicatessen products. Milk and frozen products also have a very minor affect on this customer. However, this customer is virtually unaffected by frozen products. This customer segment looks like convenience stores.

The vector [0.02 1. -0.58 -0.04 0.32 -0.62] represents a customer segment that is greatly affected by milk products, followed by groceries and delicatessen products, and then by detergents/paper products. Fresh and frozen products have almost no effect on this customer segment. This customer segment looks like cafes that require milk products to make lattes, and sell deli sandwiches and small bites to eat. For eat-in customers the cafe would require detergents/paper too.

The vector [0.02 0.01 0.03 0.03 -0.03 -1.] represents a customer segment that effectively only purchase delicatessen products. This type of customer would be deli shops, that only sell deli products to-go (no dine

in).

The vector [0.03 -0.29 1. 0.06 -0.12 -0.23] represents a customer segment that is mostly affected by grocery products, followed by milk and delicatessen, and slightly affected by detergents/paper products. This customer segment looks like grocery stores.

The vector [0.08 0.01 -0.07 -1. 0.05 0.53] represents a customer segment that is mostly affected by frozen products, followed by delicatessen products. This customer segment looks like stores that cater to busy people who do not have time to make their own meals. The store may sell frozen TV dinners, or ready-made deli sandwiches.

The vector [1. -0.22 -0.16 -0.17 0.52 -0.26] represents a customer segment that is mostly affected by fresh products, while being significantly affected by all other product categories. This customer segment looks like a farmer's market.

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:

The advantages of K-Means clustering is that it is a fast way to perform clustering (in terms of computational runtime), compared to Gaussian Mixture Models. However, K-Means clustering only provides information about which cluster a data point belongs to, but no information about the likelihood/probability of that data point belonging to said cluster.

This is in contrast to clustering using Gaussion Mixture Models. Although it runs slower than K-Means, Gaussian Mixture Models can not only provide information about which cluster a data point belongs to, it can also provide the probability that a particular data point belongs to said cluster. Also, Gaussian Mixture Models can calculate more complex decision boundaries compared to K-Means.

Since we have a small dataset for the computational power available on standard personal computers, I believe Gaussian Mixture Models will work better for our problem at hand.

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 [177]: # Import clustering modules
          from sklearn.cluster import KMeans
          from sklearn.mixture import GMM

In [178]: # TODO: First we reduce the data to two dimensions using PCA to capture variation
          pca = PCA(n_components=2)
          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]]
```

Using K-Means Clustering

This is included for completeness, to contrast the decision boundaries of K-Means versus Gaussian Mixture Models

In [179]:
```
# 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'
num_clusters = 3
clusters = KMeans(n_clusters=num_clusters).fit(reduced_data)

print clusters
```

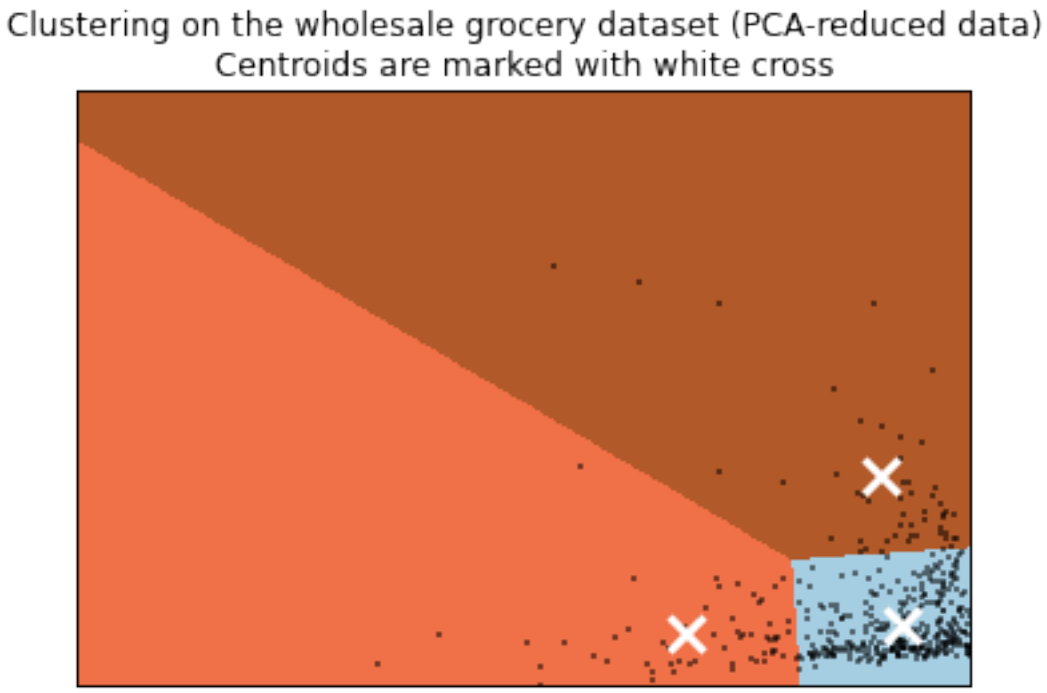
```
KMeans(copy_x=True, init='k-means++', max_iter=300, n_clusters=3, n_init=10,
    n_jobs=1, precompute_distances='auto', random_state=None, tol=0.0001,
    verbose=0)
```

In [180]:
```
# 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 [181]:
```
# TODO: Find the centroids for KMeans or the cluster means for GMM
centroids = clusters.cluster_centers_
print centroids
```

```
[[  4165.1217824   -3105.15811456]
 [-23978.86566553  -4445.56611772]
 [  1341.31124554  25261.39189714]]
```

In [182]:
```
# 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()
```



Using Gaussian Mixture Model Clustering

First, try clustering using 2 clusters, to visually see what kind of decision boundary we get

In [183]:
```
# 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'
clusters = GMM(n_components=2).fit(reduced_data)

print clusters
```

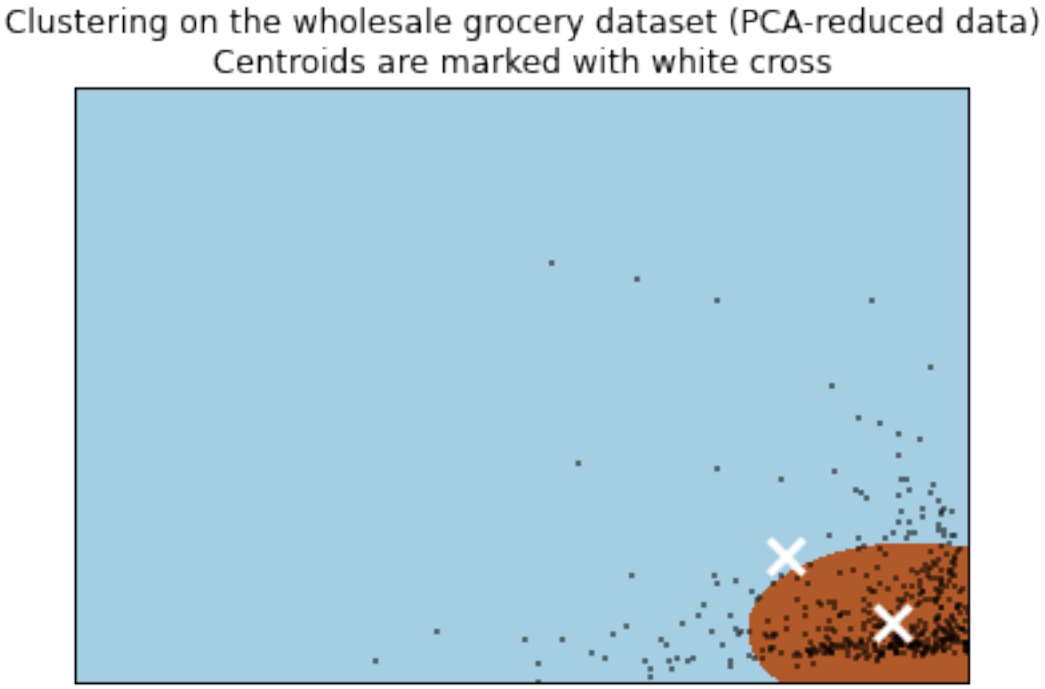
```
GMM(covariance_type='diag', init_params='wmc', min_covar=0.001,
  n_components=2, n_init=1, n_iter=100, params='wmc', random_state=None,
  thresh=None, tol=0.001)
```

In [184]:
```
# 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 [185]:
```
# TODO: Find the centroids for KMeans or the cluster means for GMM
centroids = clusters.means_
print centroids
```

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

In [186]:
```
# 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()
```



Using 2 clusters above, the clustering and decision boundary does not look right. Let's use 3 clusters and re-run our clustering algorithm

In [202]:
```
# 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'
clusters = GMM(n_components=num_clusters).fit(reduced_data)

print clusters
```

```
GMM(covariance_type='diag', init_params='wmc', min_covar=0.001,
  n_components=3, n_init=1, n_iter=100, params='wmc', random_state=None,
  thresh=None, tol=0.001)
```

In [203]:
```
# 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()])
```

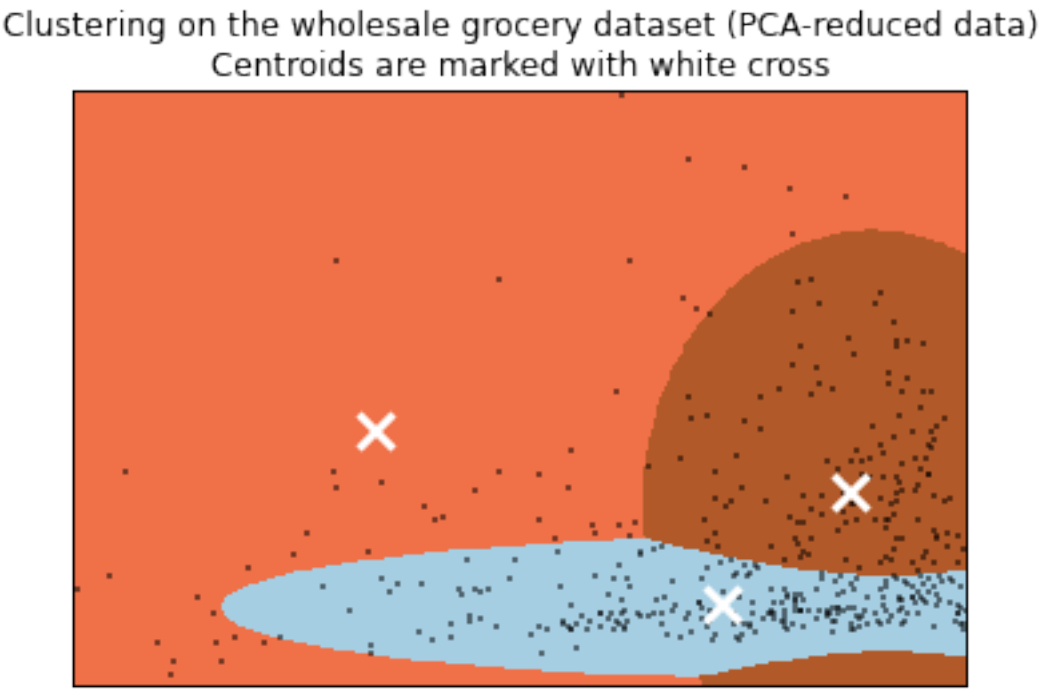
In [204]:
```
# TODO: Find the centroids for KMeans or the cluster means for GMM
centroids = clusters.means_
print centroids
```

```
[[   269.05318679  -6506.88683442]
 [-17879.18623839  10122.79246625]
 [  6987.95079141   4249.82914044]]
```

In [205]:
```
# 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()
```

Zoom in on the plot above

In [206]:
```
# Same plot as above, but zoomed-in
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 + 0.6*(x_max-x_min), x_max)
plt.ylim(y_min, y_max - 0.5*(y_max-y_min))
plt.xticks(())
plt.yticks(())
plt.show()
```



Sampling elements from each cluster and interpreting them Let's just take a look at the centroid for each of the 3 clusters:

centroid of blue cluster = [313.48539412 -6497.20521133] centroid of brown cluster = [7007.45427668 4294.01360677] centroid of orange cluster = [-17858.6536729 10050.33930164]

Looking at the centroid of the blue cluster, this looks like a customer who buys little/moderate amounts of fresh produce, and very little amounts of groceries/milk/detergents/paper.

Looking at the centroid of the brown cluster, this looks like a customer who buys moderate amounts of groceries/milk/detergents/paper, and very little fresh produce.

Looking at the centroid of the orange cluster, this looks like a customer who buys large amounts of fresh produce, and moderate amounts of groceries/milk/detergents/paper.

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

Answer:

First, note that the horizontal axis is our first principle component from PCA, which is the component dominated by the "Fresh" category. Also note the horizontal axis is negative, so datapoints on the left of the plot represent customers who placed relatively large orders of "Fresh" items. The vertical axis is our second principle component from PCA, which is a combination of "Grocery", "Milk", and "Detergents_Paper". The datapoints on the top of the plot represent customers who placed large orders of groceries/milk/detergents/paper.

Recall that "Frozen" and "Delicatessen" categories are not represented in our 2 chosen principal components, since the quantities of frozen and delicatessen products are too small compared to the other categories. This renders frozen and delicatessen categories less relevant.

Qualitatively, the blue cluster represents customers who buy very little groceries/milk/detergents/paper, and little to moderate quantities of fresh produce. The customers represented by the blue cluster appear to be restaurants.

The brown cluster represents customers who buy very little fresh produce, but moderate to large quantities of groceries/milk/detergents/paper. The customers represented by the brown cluster appear to be supermarkets.

The orange cluster represents customers not included in the blue and brown clusters, and it's hard to pinpoint an exact category of customers for the orange cluster. However, the customers in the orange category are large customers, so they are an important category to consider. This category can be referred to as "other large customers".

1.2.2 Conclusions

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

Answer:

For deciding between PCA and ICA, I felt PCA worked better for the problem at hand. For more details, see answers to questions 3 and 4.

Deciding between K-Means and Gaussian Mixture Model (GMM) clustering, I felt GMM gave more insight into the data. Looking at the plots for both K-Means and GMM clustering methods (question 6), the plot using GMM clustering made more intuitive sense, in terms of valid clustering decision boundaries. Also, computational runtime was not an issue for GMM, given the small size of our dataset and available computing power in standard personal computers. Recall one disadvantage for GMM versus K-Means is that GMM requires more computation time.

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

Answer:

As seen from our analysis using PCA and GMM clustering, we can find 3 general customer segments: restaurants, supermarkets, and "other large customers". This information can help the company design an A/B test to better evaluate the effects of any new policies.

From https://discussions.udacity.com/t/improving-the-ability-to-get-good-results-from-an-a-b-test/44142/2, "[i]n an A/B test, you split your customers into two sets, and implement a small change on one set of customers. The idea is that if your two sets of customers are very similar otherwise, the small change should account for any differences in their behavior. You can then check if there are significant changes in behavior, before deciding whether to implement the change for everyone."

For example, to test the effects of changing from regular morning delivery to bulk evening delivery, the company can run A/B tests on its 3 main customer segments separately. Starting with its restaurant customers (as determined by our clustering algorithm), the company can choose a subset of its restaurant customers to implement the evening delivery change, and evaluate the effect versus the control group which still receives morning deliveries. The company can then repeat this for supermarkets and the other large customers, to get a more precise idea of the effects of the policy change.

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

Answer:

The feature reduction and clustering work we have done can be used to aid in supervised learning tasks in the future.

For our PCA feature reduction, we have greatly reduced the feature set from 6 features down to 2 features. This allows future supervised learning algorithm applications to run much faster. Also, we would need less data to avoid the curse of dimensionality, since we have less dimensions.

For our clustering work, we discussed above how this helps with A/B testing. Thus, any future supervised learning applications based on A/B testing results will benefit from the clustering work we have performed.

As an example for applying supervised learning, let's revisit the example scenario from question 9, where we described running A/B tests on the 3 customer segments to gain more insight into the effects of changing from regular morning delivery to bulk evening delivery. After we implement the delivery change on select customers from each of the 3 categories, we can log the change in customer order volumes for all customers across each of the 3 customer segments after a set period of time (for both customers in the control group and the variation group). During this period of time when we are logging the change in order volume, there may be some macro effects affecting all customers. Thus, we need to normalize the change in order volume of the variation group versus that of the control group. Afterwards, we can build a regression model in which the independent variables are (a) customer cluster, and (b) total customer order volume. The dependent variable will be the normalized change in order volume.

Using this regression model, the company can then predict the overall revenue gain/loss from implementing the delivery schedule change given its customer profile, versus the cost savings from the delivery schedule

change. This will help determine if the delivery schedule change will ultimately have a positive or negative impact on the company's bottom line. Or, the company can decide that it only wants to implement the delivery schedule change for certain customer segments, but not others.