Robert F.K. Martin
ID 1505151
Csci 5523
21 Sep 2011
Homework 1

Problem 1: Classify the following attributes as binary, discrete, or continuous. Also classify them as qualitative (nominal or ordinal) or quantitative (interval or ratio). Some cases may have more than one interpretation, so briefly indicate your reasoning if you think there may be more ambiguity.

a) House numbers assigned for a given street: Discrete, ordinal. House numbers are typically integers, so therefore discrete. And ordinal since house numbers really only track relative locations and have no units.

b) Your calorie intake per day: Continuous, ratio. Caloric intake, although sometimes rounded to integers, is indeed a real-valued measurement. And it is meaningful to say that 2000 calories is twice as many as 1000 calories.

c) Shape of a geometric objects commonly found in geometry classes: Disrete, nominal. Names of shapes are words and words are countably infinite. And since the words are somewhat arbitrary, distinctness is the only property they exhibit. However, if we start talking about vertices of polygons, then we have a ratio attribute.

d) Routes in rock climbing: Discrete, ordinal. There are a countable number of route difficulties. And they exhibit only order; it is not meaningful to say that there is a 0.1 difference in difficulty between a 5.4 and a 5.5.

Problem 2: Decide which of the similarity measures listed in Chapter 2 would be most appropriate for the following situations and why.

1. Suppose two of your friends are numismatists (collecting coins from different countries as a hobby). You also have coins from various countries. You want to decide which friend has the most similar collection to you. Hint: You can represent each collection as a vector of length 196 of the official independent countries of the world, where the corresponding entry denotes number of coins collected for that country.

The cosine similarity would be appropriate since it often used for document simularity and frequency of words in a document are a good analog of counts of coins from a particular country, The cosine similarity, like the Jaccard, ignores zero entries(which would likely be the case for many of the 196 countries), but also takes into account non-binary vectors. However, the way the question sets up the data does not take into account different coins from the same countries. For example, I may have 5 $2 coins from Canada but my friend has a $1 coin, $2 coin, penny, quarter, and a nickel. We both have 5 coins but one could argue that his is more interesting. A better way to set up the problem would be to have a vector of all possible coins from all possible countries though the cosine similarity would still be the proper measure.

2. Suppose you measure the precipitation level in Minnesota for each zip code every

day. Similarity is to be computed between the precipitation levels in Minnesota today and same day of last month.

Since we are dealing with continuous data, Euclidean distance is the most appropriate. Even though the data is most likely sparse since it is rare for an entire state to have precipitation, zero precipitation levels are meaningful.

3. A nutritionist wants to measure the similarity between you and your friend based on following attributes: your height (in meters), weight (in pounds), your dietary requirement (in calories), and your daily activity level (low, medium, high, extreme). Note that the feature set includes continuous and discrete features.

Since we are dealing with different types of data, we'll need to compute similarity metrics separately and combine them. The first three metrics are continuous but have different scales, so we should compute their similarity one-by-one using something basic, like the L-1 norm, and scale them accordingly, perhaps by the average value for those types of data. Similarly, we can discretize the activity level so that it to is equally centered around one with an association like {low, medium, high, extreme}={0.5, 0.83, 1.17, 1.5}. We could then weight these new data points, but assuming we don't, we can now use another similarity metric like correlation or cosine to compare these new data vectors. Or we could compute the Mahalanobis distance in the vector space of the first three metrics to condense those to one metric. We could then take that metric and discretize the fourth metric and compute the Mahalanobis distance in that vector space. Note, this only works if we discretize the activity level in the same order as the nominal values.

Problem 3: Data reduction sampling, dimensionality reduction, or selecting a subset of features is necessary or useful for a wide variety of reasons, but can be problematic if information necessary to the analysis is lost in the process. The following questions explore several issues at a conceptual level.

a)i) If the event occurs at a rate of 0.001, i.e., 0.1% of the time, then what problems, if any, would you encounter in trying to estimate the rate from a single sample of size 100?

There is a very good chance that no events will happen within 100 samples. Even with 1000 samples, the rate would predict one event which still isn't informative.

a)ii) If the event occurs at a rate of 0.50, i.e., 50% of the time, then what problems, if any, would you encounter in trying to estimate the rate from a single sample of size 100?

Depending on what accuracy is needed, this may be sufficient. Also, the accuracy is also going to depend on what sort of probability distribution this event comes from(gaussian, uniform, etc.). This sample size will indeed generate events but will only give limited accuracy in predicting the rate. A gaussian distribution would more likely give a more accurate rate than a uniform distribution.

b) You are given a data set of 10,000,000 time series, each of which records the temperature of the Earth at a particular location on the surface of the Earth daily

for 10 years. The locations are arranged in a regular grid that covers the surface of the Earth. (Details of the exact nature of the grid are unimportant. The important fact is that each point has neighbors to the left and right, up and down.) Note that temperature displays considerable autocorrelation, i.e., the temperature at a given location and time is similar to that of nearby locations and times. The size of the data needs to be reduced so that you can apply your favorite data analysis algorithm. Both aggregation and sampling could be used to reduce the amount of data.

i) If you use aggregation, would you aggregate over location or time or both?

Ideally you would aggregate over time and space since there will be a high degree of correlation in both spatial and temporal domains. This will allow for the largest reduction in data.

ii) How would you use the spatial and temporal autocorrelation of temperature to guide you in aggregating the data?

We would look for spatial and temporal data that is (at least) pairwise correlated in space and time. Thinking bigger than pairwise, we would look for n x n templates in space that correlate well with n x n templates in the same space but neighboring time slices.

iii) If you use sampling, would you sample over location or time or both?

Since the time resolution is in days and there is seasonal variation in temperatures, spatial sampling would make the most sense. However, we could do equal sampling within each month or season to capture monthly or seasonal averages.

Would you perform random sampling or sample in some other way in order to best represent the data?

I would sample in a more intelligent way. There are a number of factors to take into account. Which continent are we sampling from? What latitude are we sampling from? What elevation are we at? Are we near a mountain range? Are we over a large body of water? Ideally we would like to sample equally among different biomes. Over land masses we would like to sample a little more densely, especially at the transition areas between biomes. Over the oceans, we could sample more sparsely since variation is more dependent on latitude than geography.

Would you prefer aggregation or sampling or both? (You can argue any of these as long as you support your answer.)

I would prefer both. Since the data is well correlated(or at least I assume it is based on my knowledge of weather), sampling should be able to reduce our data size without losing important values. Also, for the same reason, we could us aggregation in time and space to find contiguous areas that are similar in seasonal or monthly time scales. The more memory or processor expensive the data mining algorithm the more we want to reduce the size of our original data. If we have all

the time and memory in the world, then aggregation and sampling are of little value.

Problem 4: In order to compute similarity between two documents:

i) Each document can be represented as a vector of binary features, where each feature is a word of interest in the document. In this vector a 1 indicates the presence of the word and a 0 indicates its absence.

ii) Each document can be represented as a vector of term frequencies, which represent the frequency with which each term occurs in the document. (Details of exactly how this is computed are unimportant to this problem.)

Consider the following example in which there are two documents, D1 and D2, containing four words. Using the representation described in (i) the two documents are denoted as follows:

D1 = (1, 0, 0, 1), D2 = (1, 0, 1, 0). The Jaccard similarity in this case is 0.33.

Using the representation described in (ii) the two documents are denoted as follows:

D1 = (0.5, 0, 0, 0.5), D2 = (0.5, 0, 0.5, 0). The cosine similarity in this case is 0.5.

Provide an example of a pair of documents for which Jaccard similarity will be larger than cosine similarity

Let's make our documents from lyric snippets from Led Zeppelin and Justin Bieber. From Lef Zeppelin's D'yer Maker(D1) we have "Oh oh oh oh oh oh baby" and from Bieber's Baby(D2) we have "Baby baby baby oh". Using i) D1=(1,1) and D2=(1,1). Using ii), D1=(.857 .143) and D2=(.25 .75). J=1 and cosine=0.468.

b) Both the L1 distance and correlation are widely used to compare two time series. The most appropriate measure depends on the specific problem or domain requirements. Decide the proper measure for each of the following scenarios in climate data analysis:

i) We want to compare the temperatures of two cities for 100 days with respect to their levels.

In this case, we would probably look at correlation since we expect the temporal variation to be similar although the mean would probably be different. The L1 norm would capture the differences in the mean but would not account for the temporal variation.

ii) We want to compare the temperatures of two cities for 100 days with respect to

the trends (up and down that occur in temperature during the 100 days.

If we are looking at the difference between the temporal change of two cities, we should use the L1 norm since we don't expect cities to follow identical warming and cooling trends over a small time window. Over a large time window, we should see a similar increase or decrease in temps but probably at a different magnitude. The L1 norm will capture this.

Problem 5: For the following vectors, x and y, calculate the indicated similarity or distance measures:

Matlab code:

```
function similarity(x,y)

% Jaccard for binary vectors
if (min(x)>=0 && max(x)<=1 && min(y)>=0 && max(y)<=1),
  f11=sum(x&y);
  fxor=sum(xor(x,y));
  sj=sprintf('Jaccard:    disp(sj);
end

%cosine
sc=sprintf('cosine:  %.3f',x*y'/(norm(x)*norm(y)));
disp(sc);

%euclidean
se=sprintf('euclidean:  %.3f',norm(x-y));
disp(se);

%correlation
xmean=mean(x);
xstd=std(x);
ymean=mean(y);
ystd=std(y);
xprime=(x-xmean)/xstd;
yprime=(y-ymean)/ystd;
c=xprime*yprime'/(length(x)-1);
scorr=sprintf('correlation:  %.3f',c);
disp(scorr);
```

a) x = (0, 1, 1, 2, 2), y = (0, 2, 2, 4, 4) cosine, correlation, Euclidean

```
>> similarity([0 1 1 2 2],[0 2 2 4 4])
cosine:  1.000
euclidean:  3.162
```

```
correlation:  1.000
```

b) x = (0, 1, 0, 0, 0), y = (0, 1, 0, 0, 1) Jaccard, cosine, Euclidean, correlation

```
>> similarity([0 1 0 0 0],[0 1 0 0 1])
Jaccard:  0.500
cosine:  0.707
euclidean:  1.000
correlation:  0.612
```

c) x = (-1, -1, -1, -1), y = (1, 1, 1, 1) cosine, correlation, Euclidean

```
>> similarity([-1 -1 -1 -1],[1 1 1 1])
cosine:  -1.000
euclidean:  4.000
correlation:  NaN
```