Graphs and Networks Lecture 12

Inference on and from Graphs

Daniel A. Spielman October 8, 2013

12.1 Disclaimer

These notes are not necessarily an accurate representation of what happened in class. They are a combination of what I intended to say with what I think I said. They have not been carefully edited.

12.2 Overview

In this lecture, I am going to discuss two problems related to inference and graphs. The first is Respondent Driven Sampling [**?**]. I mostly refer you to the readings recommended on the course web page. RDS uses some very interesting ideas, but it is fundamentally flawed. Someone needs to find a better solution!

The second topic, which will occupy us for a few lectures, is that of making inferences from data on graphs. It goes by the names "regression on graphs" and "semi-supervised learning".

12.3 Regression on Graphs

We assume that we are given a graph $G = (V, E)$ along with labels on some of the vertices. The vertices for which we are given labels we will denote W, and the label we have been given for a node $w \in W$ will be $l(w)$. We assume that vertices that are connected by edges are more likely to have similar labels.

The labels could be boolean, say 0 or 1. Or, they could be real numbers. For example, if the graph is a social network graph, the labels could be 1 for regular voters and 0 for non-voters. Or, they could be the weights of people in the network (as in pounds).

The labels could also be categorical. For example, consider the k-nearest neighbor graphs that we formed from the `mnist` database. Each vertex corresponds to the image of a digit, and its label is the actual digit.

Given some of the labels, one would like to infer the rest. We will examine a proposal of Zhu, Gharamani and Lafferty [ZGL03] for how to do this. We begin by considering the case in which the labels are real numbers.

Imagine that there is a vertex $u \in V - W$ such that all of its neighbors are in W. A reasonable guess for the label of u would be the average of the labels over its neighbors. We will use the vector $\boldsymbol{x}$ to denote the labels that we guess. In this situtation, we guess

$$\boldsymbol{x}(u) = \frac{1}{d(u)} \sum_{(u,w)\in E} l(w).$$

To handle the general case, we extend this idea to every vertex in $V - W$. That is, we seek a guess for the labels of every vertex in $V - W$ so that each is the average of the labels at its neighbors. That is, we want $\boldsymbol{x} : V \to \mathbb{R}$ such that

$$\boldsymbol{x}(u) = \frac{1}{d(u)} \sum_{v:(u,v)\in E} \boldsymbol{x}(v), \qquad \text{for each } u \in V - W, \text{ and} \tag{12.1}$$

$$\boldsymbol{x}(u) = l(u) \qquad \text{for each } u \in W. \tag{12.2}$$

A function that satisfies these equations is said to be *harmonic* on W.

In the next section we show that one can always find a solution to these equations in a connected graph. In the following section we show that the solution is unique.

12.4 A solution exists

Let $l : W \to \mathbb{R}$ be an assignment of labels to the vertices in W. We wish to show that there exists a vector $\boldsymbol{x}$ that satisfies (12.1). To do this, we will begin by considering very special label functions. For $s \in W$, let δ_s be the function that is 1 at s and 0 at $W - \{s\}$. We will show that there is a vector $\boldsymbol{x}_s$ that satisfies those equations for δ_s.

As

$$l = \sum_{s\in W} l(s)\delta_s,$$

and the equations (12.1) are linear, this implies that

$$\sum_{s\in W} l(s)\boldsymbol{x}_s$$

satisfies the equations for l.

We now return to proving the existance of $\boldsymbol{x}_s$. Let $T = W - \{s\}$. Consider a random walk that starts at a vertex u, and stops when it finally reaches s or T. Let $F(u)$ be the probability that it hits s before t. It should be clear that F is well-defined, and that by definition $F(s) = 1$ and $F(t) = 0$ for $t \in T$.

Let's see what happens in between. For a vertex u that is neither s nor t, the probability that it steps to a neighbor v is $1/d(u)$. We have

$$\begin{aligned} F(u) &= \sum_{v:(u,v)\in E} \mathbf{Pr}\left[\text{the walk goes to } v \text{ after } u, \text{ and eventually stops at } s\right] \\ &= \sum_{v:(u,v)\in E} \mathbf{Pr}\left[\text{the walk goes to } v \text{ after } u\right] F(v) \\ &= \frac{1}{d(u)} \sum_{v:(u,v)\in E} F(v). \end{aligned}$$

So, F satisfies the equations (12.1) for δ_s.

12.5 The solution is unique

Note: we did not get to this in class.

Lemma 12.5.1. *Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two functions that are both harmonic on $V - W$ and such that for all $w \in W$*

$$\boldsymbol{x}(w) = \boldsymbol{y}(w).$$

Then, for all $u \in V - W$,

$$\boldsymbol{x}(u) = \boldsymbol{y}(u).$$

Proof. Let $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{y}$. So, $\boldsymbol{z}(w) = 0$ for all $w \in W$, and we need to show that $\boldsymbol{z}(u) = 0$ for all $u \in V$.

Let v be such that $\boldsymbol{z}(u) \leq \boldsymbol{z}(v)$ for all $u \in V$. We have

$$\boldsymbol{z}(v) = \frac{1}{d(v)} \sum_{u:(v,u)\in E} \boldsymbol{z}(u) w_{u,v}.$$

The right-hand side is the average of the values $\boldsymbol{z}(u)$ for $(u, v) \in E$. As it is also equal to the maximum of $\boldsymbol{z}$, all the values $\boldsymbol{z}(u)$ must be equal. So, $\boldsymbol{z}(u) = \boldsymbol{z}(v)$ for every neighbor u of v. By induction on paths in the graph, $\boldsymbol{z}(u) = \boldsymbol{z}(v)$ for every vertex u reachable from v. As this holds for some vertex $w \in W$, we have $\boldsymbol{z}(v) = \boldsymbol{z}(w) = 0$, and $\boldsymbol{z}(u) = 0$ for all vertices u that are reachable from v, which is all of V provided that the graph is connected. □

12.6 Experiments

Let's try experiments on the `mnist` data set. Recall that each vertex in this data set corresponded to a 28-by-28 image of a digit. We then treated these images as vectors, and created a 3-nearest-neighbor graph on them.

In class, we tried to identify the 6s. We began by creating a vector, called `l6`, that is one at the 6s and 0 everywhere else. We then choose 100 random vertices, and told our algorithm the labels of those nodes.

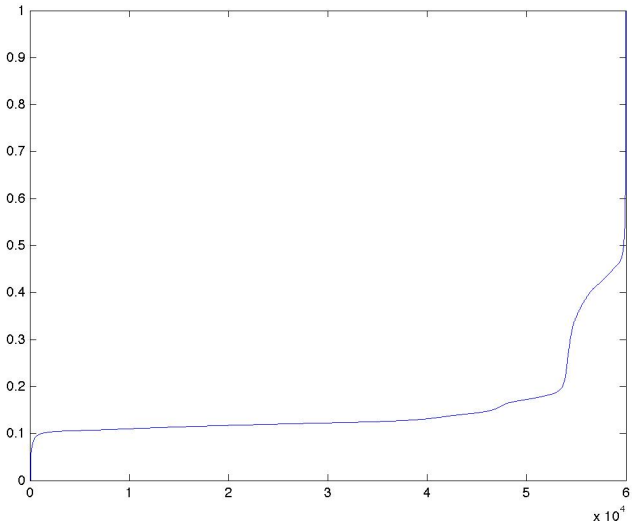
```
load mnist;
n = size(a3,1);
l6 = (labels == 6);
sum(l6)/n

ans =

     0.098633

p = randperm(n);
where = p(1:100);
sum(l6(where))

ans =

    17
```

I will then make a matrix with the harmonic equations for the rows not in `where`, and that forces the labels on the row in `where`. I also set up the left-hand side of the equations. This vector, `b`, should be zero outside of where, and agree with `l6` inside where. We will let `x` be the solution to the equations.

```
>> m = diag(sum(a3)) - a3;
>> m(where,:) = 0;
>> m(where,where) = eye(length(where));
>> b = zeros(n,1);
>> b(where) = l6(where);
>> x = m $\$ b;
```

If we now look at a plot of the values of `x`, we see that most of them are around 0.1, and the threshold for those we want is probably around 0.5.

```
>> plot(sort(x))
```

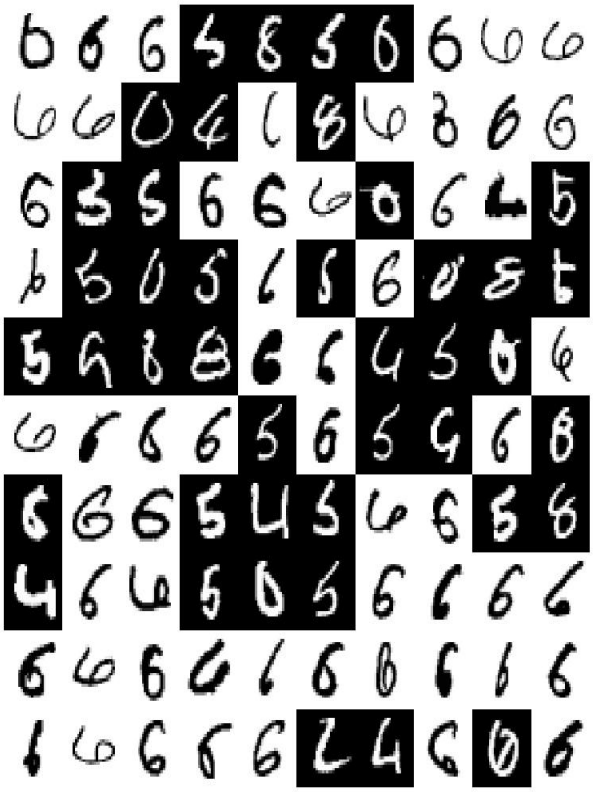


Instead of choosing the threshold numerically, I will apply our procedure from the lectures on random walks: I will chose prefix of the values in x that are largest that has the lowest conductance, and guess that those are the cluster of 6s. We see that this gives us 5890 vertices, which the number that are actually 6s is 5918. We have mis-classified 270 vertices.

```
>> st = sparsecut(a3,-x);
>> nnz(st)

ans =

        5890

>> sum(l6)

ans =

        5918

>> xr = zeros(n,1); xr(st) = 1;
>> sum(l6 ~= xr)

ans =

   270
```

I will now plot those we mis-classify. Those that are labeled 6 but which we do not recognize have a white background (note that I did this incorrectly in class).

```
>> diff = find(xr~=l6);
>> for i = 1:10, for j = 1:10,
if (l6(diff(i+(j-1)*10))),
badimg((i-1)*28+[1:28],(j-1)*28+[1:28]) = 1-imgs(:,:,diff(i+(j-1)*10));
else
badimg((i-1)*28+[1:28],(j-1)*28+[1:28]) = imgs(:,:,diff(i+(j-1)*10));
end;
end; end;
imshow(badimg)
```



References

[ZGL03] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proc. 20th Int. Conf. on Mach. Learn.*, 2003.