1 Clustering with Sample Techniques

Consider the k-center clustering problem: given a set of n points, find a set of k cluster centers with minimum radius. The *radius* of a solution is the maximum distance from a point the closest center. Finding the optimal solution R for a given instance is NP-hard. However, we can use the polynomial-time 2-approximation algorithm seen in Lecture 12 to find a solution whose radius is no greater than $2R$ for the same instance.

We are now interested in the following situation. Instead of applying our algorithm to all n points, we apply it to a random sample of size $O((k \log n)/\epsilon)$. Since usually $k \ll n$, this tends to be a small fraction of the input points. Dealing with the sample should be much cheaper than dealing with the whole set.

If the optimal solution for the original set is R, it is easy to see that the solution of the 2-approximate algorithm on the sample will have value no greater than $2R$. One only has to note that the optimal solution of the sample has radius no greater than R, as long as we do not require cluster centers also belong to the set of points. Since the proof shown in Lecture 12 does not assume the presence of the cluster centers, it still holds in this case.

Even though we can find a valid solution to the sample with radius no greater than $2R$, it may not cover some of the points in the original instance. Fortunately, as seen in the previous lecture, with high probability the fraction of points that are not covered by the solution is small, at most ϵ. If more than an ϵ fraction of the points were not covered, with high probability one of them would be in the sample (and, therefore, covered).

Therefore, we have proven that, given an instance with n points of radius R, we can sample just $O((k \log n)/\epsilon)$ points and still find (in polynomial time) a solution a solution with radius $2R$ that with high probability will cover at least a $1 - \epsilon$ fraction of the points. Note that our analysis relies on some properties of the 2-approximate algorithm we are using to find a solution to the sample. If we used an exact algorithm instead, the nice properties found in this case would not necessarily carry on.

2 Ranking Web Pages

The amount of information available on the Web is huge. However, it is often the case that we need information on a particular topic. We cannot even hope to be able to read all pages to get this information. Instead, it would be interested if we could automatically find the pages that are most relevant to our query. That is the main task of search engines, such as Google.

To determine which pages are the most relevant, we need some objective method to *rank* those pages. This section presents two such methods, *PageRank* and *Hubs and Authorities*. As we will see, PageRank is basically query-indepedent, whereas Hubs and Authorities depends heavily on the subject we are interested in.

Both methods rely on the link structure of the Web to rank pages. If a page A contains a certain page B, we can infer that the author of A for some reason thinks B is relevant. If many pages point to B, a reasonable conclusion is that B must be an important page, and therefore should have a high rank. Of course, people can be (and often are) malicious: the same person can create several pages whose only purpose is to link to some other page just to make it look relevant. For most of this section, we will assume a more "honest" model for the Web. It should be clear though that real-life implementations of the methods presented here must take human nature into account, or else they risk being useless in practice.

As an aside, it is important to notice that the idea of examining the link structure of a graph is not used exclusively to rank web pages. It is often used in citation analysis: if an academic paper is cited a lot, it is probably useful and contains important information. Similar concepts are often used in social networks to analyze interrelationships in social groups.

2.1 PageRank

An important aspect of PageRank is that it is query-independent. All pages on the web are ranked on their "intrinsic" value, regardless of topic. Whenever a query is made, PageRank must be combined with query-specific measures to determine the relative importance in a given context.

Let v be some web page, and let B_v be the set of pages that contain links to v, and let F_v be the set of pages v points to, with $|F_v| = N_v$. The rank of v is denoted by $R(v)$. The higher the rank, the more relevant the page is. According to PageRank, if a well-ranked page u points to page v, page v must have some importance. In a sense, u contributes to increase the rank of v. In a first approximation, we could recursively define the PageRank of v as follows:

$$R(v) = \sum_{u \in B_v} \frac{R(u)}{N_u}. \tag{1}$$

This equation shows that each page u that points to v contributes with some of its rank. More precisely, the rank of a given page u is split evenly among all pages it points to (as the term $R(u)/N_u$ shows). This is a recursive definition; the actual rank of the pages would be calculated iteratively.

There is a problem, though. It would be desirable (although not strictly necessary) to preserve the total rank during the iterative calculation. The total rank is just the sum of the left-hand side of equation 1 for all existing pages (represented by P):

$$\text{LHS} = \sum_{v \in P} R(v),$$

If we instead choose to sum the right-hand sides, we get a similar expression:

$$\text{RHS} = \sum_{u \in P'} R(u)$$

The problem is that the set P' over which the sum is taken in the latter expression is *not* equal to P, the complete set of pages. P' is restricted to pages that have links to other pages. This would cause the total rank to decrease from one iteration to another. To avoid this problem, we add a normalization factor c:

$$R(v) = c \sum_{u \in B_v} \frac{R(u)}{N_u}. \tag{2}$$

There are other problems that are not addressed by this expression. Consider a situation in which two pages point to each other (and to nowhere else). If there is a link from the "rest" of the Web (assume it is biconnected) to this little cluster, it will tend to get all the available rank to itself. To minimize this problem, we can require that each page has a minimum rank. This is captured by the following expression:

$$R(v) = c \sum_{u \in B_v} \frac{R(u)}{N_u} + E(v). \tag{3}$$

The new vector E can be seen as a "source" of PageRank. The next section presents an alternative way to interpret its entries.

2.1.1 The Random Surfer Model

Consider the following model for a "typical" person surfing the Web. After reading a certain page, the person may decide, with probability α, to go to some (random) other page on the Web (by explicitly typing its URL, for instance). With probability $1 - \alpha$, the surfer follows one of the outgoing links of the page, selected at random and uniformly.

What is the probability $p(v)$ that the random surfer is at page v at any given moment? It's easy to see that it is

$$p(v) = \alpha \cdot e(v) + (1 - \alpha) \cdot \sum_{u \in B_v} \frac{p(u)}{N_u}.$$

As before, B_v represents the set of pages that point to v, and N_u represents the number of links in page u. In this case e is the vector of *reset probabilities*; for each page, it contains the probability that this page will be selected when the surfer performs a *random jump* (i.e., when she doesn't follow a link).

This expression is remarkably similar to Equation 3, which indicates that this *Random* Surfer Model just provides a different way to look at PageRank. Although this model does not capture in full the behavior of a "real" surfer, it does provide some insight into why PageRank should work reasonably well.

2.1.2 PageRank in Practice

PageRank was introduced by Larry Page et al. in "The PageRank Citation Ranking: Bringing Order to the Web" (Stanford Digital Library Working Papers, 1998). The search engine presented in that paper would later become Google.

In the paper, it is suggested that defining $e(v)$ to be the same for all pages on the Web is a reasonable approach. However, the author also looked at what would happen if some pages were given greater initial rank than others. In their experiments, two possible scenarios were considered. In the first one, Netscape's Home Page (a relevant commercial site) was given a reset probability of 1, and all other pages were given zero to start with. The second scenario was similar, but John McCarthy's home page at Stanford was given the highest reset probability.

As expected, these variations very noticeable local effects. Home pages of faculty members at the Stanford's Department of Computer Science had a much higher PageRank in the second scenario. However, these changes have a global effect that is still perceptible. Academic pages in general (as opposed to commercial pages) were ranked higher when John McCarthy's was assigned a higher reset probability.

This experiment shows that it is possible — at least in theory — to customize PageRank to fit individual interests. By adjusting the reset probabilities (the values in e), a personalized set of ranks could be create. In practice, however, creating a different set of ranks for each individual is prohibitive, given the iterative and global nature of PageRank calculations. A less ambitious approach, such as creating static categories, may be feasible instead.

2.2 Hubs and Authorities

We now discuss another ranking strategy. Instead of globally ranking pages, this methods assigns ranks that are specific to query we are interested in. This method, for reasons that we become clear soon, is called *Hubs and Authorities*, and was proposed by Jon Kleinberg in the paper "Authoritative Sources in a Hyperlinked Environment" (presented at SODA'98).

The basic idea is as follows. First, submit a query to a search engine (in the original paper, tests were made with Altavista). The query could be "automibile manufacturers", for instance. This will return several pages in which this string occurs. However, note that this may not include obvious candidates such as GM or Ford, simply because that particular expression may not appear in their Web pages. Pick the top 200 pages returned by the query, together with at most 50 of the pages they point to, and 50 of the pages that point to them. Together, these pages induce a relatively small subgraph $G = (V, E)$ of the Web graph, where V is the set of pages ($|V| \leq 300$) and E is the set of (directed) links.

The basic idea is to look at the structure of the links, trying to determine the relative importance of those 300 pages. Intuitively, a page should be an *authority* on the subject matter determined by the query — i.e., a page with relevant information — if there are several pages pointing to it. Some other pages may have very limited specific information on the subject, but contain links to several authorities. Such pages are called *hubs*. Of course, it is possible for a single page to be both an authority and a hub.

We can further refine our definition as follows: an authority is a page that is linked to by lots of hubs, and a hub is a page that points to lots authority. Although these definition have a circular "flavor", they can in fact be translated into an iterative computation process.

We associate to each page p a pair (x_p, y_p) of values, where x_p is the authority score and y_p is the hub score. These values are initially the same for all pages, and are further refined by two operations. An I ("input") operation, consists of recomputing the values of x_p as follows:

$$x_p = \sum_{(q,p) \in E} y_p.$$

This means simply that the authority score of each page p will be recomputed as the sum of the hub scores of all pages that point to it. Then we perform an O ("output") operation:

$$y_p = \sum_{(p,q) \in E} x_p.$$

The hub score of a page p is recalculated as the sum of the authority scores of all pages p has links to.

After several iterations (alternating between these definitions), each page will have scores that actually represent its relative rank as a hub and an an authority. Typically, reporting the pages with the top 5 to 10 authority scores and the pages with the top 5 to 10 hub scores is a reasonable strategy.

Note that the underlying assumption made by this method is that a link of page A to page B reflects the fact that the author of page A confers a certain degree of authority to page B. This sounds reasonable, but one must be careful. A large portion of the links found on the Web are purely navigational ("Click here to return to the main menu"). To improve the quality of the results provided by the algorithm, it is suggested that only cross-domain links are considered when G is built.