On Clustering Using Random Walks

David Harel and Yehuda Koren

Dept. of Computer Science and Applied Mathematics
The Weizmann Institute of Science, Rehovot, Israel
{harel,yehuda}@wisdom.weizmann.ac.il

Abstract. We propose a novel approach to clustering, based on deterministic analysis of random walks on the weighted graph associated with the clustering problem. The method is centered around what we shall call *separating operators*, which are applied repeatedly to sharpen the distinction between the weights of inter-cluster edges (the so-called separators), and those of intra-cluster edges. These operators can be used as a stand-alone for some problems, but become particularly powerful when embedded in a classical multi-scale framework and/or enhanced by other known techniques, such as agglomerative clustering. The resulting algorithms are simple, fast and general, and appear to have many useful applications.

1 Introduction

Clustering is a classical problem, applicable to a wide variety of areas. It calls for discovering natural groups in data sets, and identifying abstract structures that might reside there. Clustering methods have been used in computer vision [11,2], VLSI design [4], data mining [3], web page clustering , and gene expression analysis.

Prior literature on the clustering problem is huge, see e.g., [7]. However, to a large extent the problem remains elusive, and there is still a dire need for a clustering method that is natural and robust, yet very efficient in dealing with large data sets.

In this paper, we present a new set of clustering algorithms, based on deterministic exploration of random walks on the weighted graph associated with the data to be clustered. We use the similarity matrix of the data set, so no explicit representation of the coordinates of the data-points is needed. The heart of the method is in what we shall be calling *separating operators*, which are applied to the graph iteratively. Their effect is to 'sharpen' the distinction between the weights of inter-cluster edges (those that ought to separate clusters) and intra-cluster edges (those that ought to remain inside a single cluster), by decreasing the former and increasing the latter. The operators can be used on their own for some kinds of problems, but their power becomes more apparent when embedded in a classical multi-scale framework and when enhanced by other known techniques, such as agglomerative or hierarchical clustering.

The resulting algorithms are simple, fast and general. As to the quality of the clustering, we exhibit encouraging results of applying these algorithms to several

R. Hariharan, M. Mukund, and V. Vinay (Eds.): FSTTCS 2001, LNCS 2245, pp. 18–41, 2001.

recently published data sets. However, in order to be able to better assess its usefulness, we are in the process of experimenting in other areas of application too.

2 Basic Notions

We use standard graph-theoretic notions. Specifically, let $G(V, E, w)$ be a weighted graph, which should be viewed as modeling a relation E over a set V of entities. Assume, without loss of generality, that the set of nodes V is $\{1, \ldots, n\}$. The w is a weighting function $w : E \longrightarrow \mathbb{R}^+$, that measures the similarity between pairs of items (a higher value means more similar). Let $S \subseteq V$. The set of nodes that are connected to some node of S by a path with at most k edges is denoted by $V^k(S)$. The degree of G, denoted by $deg(G)$, is the maximal number of edges incident to some single node of G. The subgraph of G induced by S is denoted by $G(S)$. The edge between i and j is denoted by $\langle i, j \rangle$. Sometimes, when the context is clear, we will write simply $\langle i, j \rangle$ instead of $\langle i, j \rangle \in E$.

A *random walk* is a natural stochastic process on graphs. Given a graph and a start node, we select a neighbor of the node at random, and 'go there', after which we continue the random walk from the newly chosen node. The probability of a transition from node i to node j, is

$$p_{ij} = \frac{w(i,j)}{d_i}$$

where $d_i = \sum_{\langle i,k \rangle} w(i,k)$ is the *weighted degree* of node i.

Given a weighted graph $G(V, E, w)$, the associated *transition matrix*, denoted by M^G, is the $n \times n$ matrix in which, if i and j are connected, the (i, j)'th entry is simply p_{ij}. Hence, we have

$$M^G_{ij} = \begin{cases} p_{ij} & \langle i, j \rangle \in E \\ 0 & \text{otherwise} \end{cases}$$

Now, denote by $P^k_{visit}(i) \in \mathbb{R}^n$ the vector whose j-th component is the probability that a random walk originating at i will visit node j in its k-th step. Thus, $P^k_{visit}(i)$ is the i-th row in the matrix $(M^G)^k$, the k'th power of M^G.

The *stationary distribution* of G is a vector $p \in \mathbb{R}^n$ such that $p \cdot M^G = p$. An important property of the stationary distribution is that if G is non-bipartite, then $P^k_{visit}(i)$ tends to the stationary distribution as k goes to ∞, regardless of the choice of i.

The *escape probability* from a source node s to a target node t, denoted by $P_{escape}(s, t)$, is defined as the probability that a random walk originating at s will reach t before returning to s. This probability can be computed as follows. For every $i \in V$, define a variable ρ_i satisfying:

$$\rho_s = 0, \qquad \rho_t = 1, \quad \text{and}$$
$$\rho_i = \sum_{\langle i,j \rangle} p_{ij} \cdot \rho_j \qquad \text{for } i \neq s,\ i \neq t$$