Replicator Graph Clustering

Michael Donoser
http://vh.icg.tugraz.at

Institute for Computer Graphics and Vision
Graz University of Technology, Austria

Motivation Data clustering, which aims at identifying a varying number of clusters and unique cluster assignments in a fully unsupervised manner, is an important topic in machine learning and computer vision. Many of the available methods in this field such as k-means or mean shift are based on a Euclidean assumption. In this work we overcome the shortcomings of such an assumption by considering the underlying similarity manifold using diffusion processes, which allows to handle non-metric data. Core idea of our novel approach is to combine an effective diffusion process, based on iteratively approaching evolutionary stable strategies from the field of game theory, with a provably optimal clustering step that analyzes a specific graph structure, that we denote as *Replicator Graph*.

Our clustering method (*Replicator Graph Clustering*) belongs to the field of pairwise or proximity-based based clustering approaches assuming that the input is an $N \times N$ affinity matrix $\mathbf{A} = (a_{ij})$, where each entry a_{ij} measures the similarity between two specific data points-to-be-clustered i and j. The goal of clustering is to uniquely assign each of the N data points to one of a set of clusters $\mathcal{C} = (\mathcal{C}_1, \mathcal{C}_2, \ldots \mathcal{C}_C)$, where C is an automatically found number of clusters. Our method mainly consists of three subsequent steps (a) diffusing affinities by finding personalized evolutionary stable strategies of non-cooperative games (b) building a mutual k-nearest neighbor graph representing the underlying manifold and (c) applying a graph based clustering strategy which identifies the final clusters. Individual steps have low computational complexity which leads to an efficient clustering method, scaling well with an increasing number of data points.

Diffusing Affinities by Game Theory The first step of our method is to diffuse the provided affinities considering the underlying data manifold. For example, in [1] it was shown that such diffusion processes are able to significantly improve retrieval performance, and we aim at exploiting this property for our task of clustering. Our main idea is to consider each data point independently and to diffuse the affinities in relation to the current query data point, similar as in related semi-supervised learning [2] tasks. Query-specific diffusion means, that we aim at converting the query specific similarities $A_{i.}$ (the i-th row of the given affinity matrix $\mathbf{A}$) into a new $N \times 1$ vector representation $\mathbf{x}^*$ by maximizing its agreement to the underlying similarity manifold spanned by the affinity matrix $\mathbf{A}$. For query-specific diffusion we adapt evolutionary dynamics from the field of game theory [4] to our specific requirements. Finally, all query-specific diffused affinities stacked together build the updated affinity matrix $\mathbf{A}^*$, which considers the similarity distribution on the underlying manifold. Algorithm 1 summarizes the diffusion process.

Algorithm 1: Algorithm for Affinity Diffusion

Input: $N \times N$ Affinity matrix $\mathbf{A}$ and number of neighbors S
Output: Updated $N \times N$ Affinity matrix $\mathbf{A}^*$

$\mathbf{A}^* = []$
Calculate S-nearest neighbors per query and build $\mathbf{A}_{SNN}$
for $i = 1$ **to** N **do**
  Apply replicator dynamics to $\mathbf{A}_{SNN}$ using $\overline{\mathbf{A}}_{i.}$ as initialization and obtain evolutionary stable strategy $\mathbf{x}^{i*}$
  $\mathbf{A}^* = \begin{bmatrix} \mathbf{A}^* \\ [\mathbf{x}^{i*}]^T \end{bmatrix}$
return $\mathbf{A}^*$

Building Graph Structure and Clustering Based on the diffused affinity matrix $\mathbf{A}^*$, we build a specific graph structure denoted as mutual *kNN* graph (*mkNN*), which is surprisingly seldom used despite its interesting properties. The *mkNN* adds an additional constraint to the *kNN* graph, which requires that two connected data points belong to each others k-neighborhood, i. e. $j \in k\text{NN}(i)$ and $i \in k\text{NN}(j)$. In such a way we obtain our *Replicator Graph* by building the *mkNN* graph structure for the affinity matrix $\mathbf{A}^*$. Clustering is then done by analyzing th obtained *Replicator Graph* using a segmentation method proposed in [3], which precedes by merging clusters in decreasing order of the edge weights separating them. This is efficiently done by a variant of a minimum spanning tree method. The iterative merging stops once the final clustering is neither too fine (cluster should be merged due to high similarity) nor too coarse (cluster should be split due to boundary evidence). In such a way the algorithm obeys the global properties of being neither too fine nor too coarse and returns a global optimal (!) solution, although the algorithm only makes greedy decisions. For more details on the algorithm and proofs of the global optimality, see [3].

Experiments Experiments first analyze the influence of individual parameters and then demonstrate the efficacy of combining diffusion with the clustering approach. As final experiment we compare our proposed method on several data sets to related methods. We analyze three different data sets: (1) MPEG-7 CE Shape-1 Part-B (shape silhouettes, 1400 instances, 70 classes), (2) CALTECH 101 (object categories, 1010 instances, 101 clusters) and (3) UNIPEN (handwritten letters, 250 instances, 5 classes). We fixed the parameters for our method to $S = 35$ and $K = 10$. Table 1 shows results of the Normalized Mutual Information (NMI) score and runtimes on these data sets. As can be seen our proposed method provides competitive clustering results on all data sets in short computation time, e. g. being significantly faster than affinity propagation. Huge improvements against the k-Means baseline (+8.27% on MPEG-7, +15.21% on CALTECH 101 and +29.07% on UNIPEN) demonstrate the importance of considering the underlying manifold for clustering, which is implicitly done by the diffusion scheme.

Data	K-Means	Spectral	Aff.Prop	Proposed
MPEG	89.19 (**1.1s**)	95.99 (3.5s)	91.11 (14s)	**97.46** (2.7s)
CALT	41.19 (2.2s)	55.75 (1.3s)	55.65 (6.0s)	**56.40** (**0.7s**)
UNIP	42.59 (**0.1s**)	68.49 (0.1s)	59.58 (0.5s)	**71.66** (0.2s)

Table 1: Comparison of clustering quality (normalized mutual information – the higher the better) and runtime (in seconds) on several data sets.

Properties Our proposed clustering method has several important properties. First, we do not have any restrictions (like affinities have to satisfy metric properties) on the affinity matrix, e. g. negative and asymmetric affinity matrices can be handled. Second the diffusion step improves pairwise affinities due to considering the underlying manifold yielding improved clustering performance. Third, we are able to apply a provably optimal clustering method, which analyzes the widespread cluster criterium of high internal coherency and external incoherency. Fourth, we automatically identify the number of clusters using a single parameter to influence cluster granularity. Finally, we achieve improved clustering quality in comparison to state-of-the-art methods at low computation time in application fields like clustering object category images, handwritten letters and shape silhouettes.

[1] M. Donoser and H. Bischof. Diffusion processes for retrieval revisited. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[2] A. Erdem and M. Pelillo. Graph transduction as a noncooperative game. *Neural Computation*, 24(3):700–723, 2012.

[3] P. Felzenszwalb and F. Huttenlocher. Efficient graph-based image segmentation. *Intern. Journal of Computer Vision*, 59(2):167–181, 2004.

[4] P. Taylor and Jonker L.B. Evolutionarily stable strategies and game dynamics. *Mathematical Biosciences*, 40, 1978.