SUBCLU

From Wikipedia, the free encyclopedia

This article **needs attention from an expert in Statistics**. Please add a *reason* or a *talk* parameter to this template to explain the issue with the article. WikiProject Statistics (or its Portal) may be able to help recruit an expert. *(February 2010)*

This article **relies largely or entirely upon a single source**. Relevant discussion may be found on the talk page. Please help improve this article by introducing citations to additional sources. *(February 2010)*

SUBCLU is an algorithm for clustering high-dimensional data by Karin Kailing, Hans-Peter Kriegel and Peer Kröger.[1] It is a subspace clustering algorithm that builds on the density-based clustering algorithm DBSCAN. SUBCLU can find clusters in axis-parallel subspaces, and uses a bottom-up, greedy strategy to remain efficient.

Approach [edit]

SUBCLU uses a monotonicity criteria: if a cluster is found in a subspace S, then each subspace $T \subseteq S$ also contains a cluster. However, a cluster $C \subseteq DB$ in subspace S is not necessarily a cluster in $T \subseteq S$, since clusters are required to be maximal, and more objects might be contained in the cluster in T that contains C. However, a density-connected set in a subspace S is also a density-connected set in $T \subseteq S$.

This *downward-closure property* is utilized by SUBCLU in a way similar to the Apriori algorithm: first, all 1-dimensional subspaces are clustered. All clusters in a higher-dimensional subspace will be subsets of the clusters detected in this first clustering. SUBCLU hence recursively produces $k+1$-dimensional candidate subspaces by combining k-dimensional subspaces with clusters sharing $k-1$ attributes. After pruning irrelevant candidates, DBSCAN is applied to the candidate subspace to find out if it still contains clusters. If it does, the candidate subspace is used for the next combination of subspaces. In order to improve the runtime of DBSCAN, only the points known to belong to clusters in one k-dimensional subspace (which is chosen to contain as little clusters as possible) are considered. Due to the downward-closure property, other point cannot be part of a $k+1$-dimensional cluster anyway.

Pseudocode [edit]

SUBCLU takes two parameters, ϵ and $MinPts$, which serve the same role as in DBSCAN. In a first step, DBSCAN is used to find 1D-clusters in each subspace spanned by a single attribute:

$$
\begin{aligned}
&SUBCLU(DB, eps, MinPts)\\
&\quad S_1 := \emptyset\\
&\quad C_1 := \emptyset\\
&\quad for\ each\ a \in Attributes\\
&\quad\quad C^{\{a\}} = DBSCAN(DB, \{a\}, eps, MinPts)\\
&\quad\quad if(C^{\{a\}} \neq \emptyset)\\
&\quad\quad\quad S_1 := S_1 \cup \{a\}\\
&\quad\quad\quad C_1 := C_1 \cup C^{\{a\}}\\
&\quad\quad endif\\
&\quad end\ for
\end{aligned}
$$

In a second step, $k+1$-dimensional clusters are built from k-dimensional ones:

$$
\begin{aligned}
&k := 1 \\
&while(C_k \neq \emptyset) \\
&\quad CandS_{k+1} := GenerateCandidateSubspaces(S_k) \\
&\quad for\,each\,cand \in CandS_{k+1} \\
&\quad\quad bestSubspace := \min_{s \in S_k \wedge s \subset cand} \sum_{C_i \in C^s} |C_i| \\
&\quad\quad C^{cand} := \emptyset \\
&\quad\quad for\,each\,cluster\,cl \in C^{bestSubspace} \\
&\quad\quad\quad C^{cand} := C^{cand} \cup DBSCAN(cl, cand, eps, MinPts) \\
&\quad\quad\quad if\,(C^{cand} \neq \emptyset) \\
&\quad\quad\quad\quad S_{k+1} := S_{k+1} \cup cand \\
&\quad\quad\quad\quad C_{k+1} := C_{k+1} \cup C^{cand} \\
&\quad\quad\quad end\,if \\
&\quad\quad end\,for \\
&\quad end\,for \\
&\quad k := k+1 \\
&end while \\
&end
\end{aligned}
$$

The set S_k contains all the k-dimensional subspaces that are known to contain clusters. The set C_k contains the sets of clusters found in the subspaces. The $bestSubspace$ is chosen to minimize the runs of DBSCAN (and the number of points that need to be considered in each run) for finding the clusters in the candidate subspaces.

Candidate subspaces are generated much alike the Apriori algorithm generates the frequent itemset candidates: Pairs of the k-dimensional subspaces are compared, and if they differ in one attribute only, they form a $k+1$-dimensional candidate. However, a number of irrelevant candidates are found as well; they contain a k-dimensional subspace that does not contain a cluster. Hence, these candidates are removed in a second step:

$$
\begin{aligned}
&GenerateCandidateSubspaces(S_k) \\
&\quad CandS_{k+1} := \emptyset \\
&\quad for\,each\,s_1 \in S_k \\
&\quad\quad for\,each\,s_2 \in S_k \\
&\quad\quad\quad if\,(s_1\,and\,s_2\,differ\,in\,exactely\,one\,attribute) \\
&\quad\quad\quad\quad CandS_{k+1} := CandS_{k+1} \cup \{s_1 \cup s_2\} \\
&\quad\quad\quad end\,if \\
&\quad\quad end\,for \\
&\quad end\,for \\
&\quad \textit{// Pruning of irrelevant candidate subspaces} \\
&\quad for\,each\,cand \in CandS_{k+1} \\
&\quad\quad for\,each\,k-element\,s \subset cand \\
&\quad\quad\quad if\,(s \notin S_k) \\
&\quad\quad\quad\quad CandS_{k+1} = CandS_{k+1} \setminus \{cand\} \\
&\quad\quad\quad end\,if \\
&\quad\quad end\,for \\
&\quad end\,for \\
&end
\end{aligned}
$$

Availability [edit]

An example implementation of SUBCLU is available in the ELKI framework.

References [edit]

1. ^ Karin Kailing, Hans-Peter Kriegel and Peer Kröger. *Density-Connected Subspace Clustering for High-Dimensional Data*. In: *Proc. SIAM Int. Conf. on Data Mining (SDM'04)*, pp. 246-257, 2004.

Categories: Data clustering algorithms

This page was last modified on 10 January 2013, at 19:37.

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.

Privacy policy About Wikipedia Disclaimers Contact Wikipedia Developers Mobile view