OPTICS algorithm

From Wikipedia, the free encyclopedia

Ordering points to identify the clustering structure (**OPTICS**) is an algorithm for finding density-based clusters in spatial data. It was presented by Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel and Jörg Sander.[1] Its basic idea is similar to DBSCAN,[2] but it addresses one of DBSCAN's major weaknesses: the problem of detecting meaningful clusters in data of varying density. In order to do so, the points of the database are (linearly) ordered such that points which are spatially closest become neighbors in the ordering. Additionally, a special distance is stored for each point that represents the density that needs to be accepted for a cluster in order to have both points belong to the same cluster. This is represented as a dendrogram.

Basic idea [edit]

Like DBSCAN, OPTICS requires two parameters: ε, which describes the maximum distance (radius) to consider, and $MinPts$, describing the number of points required to form a cluster. A point p is a *core point* if at least $MinPts$ points are found within its ε-neighborhood $N_\varepsilon(p)$. Contrary to DBSCAN, OPTICS also considers points that are part of a more densely packed cluster, so each point is assigned a *core distance* that describes the distance to the $MinPts$th closest point:

$$\text{core-dist}_{\varepsilon,MinPts}(p) = \begin{cases} \text{UNDEFINED} & \text{if } |N_\varepsilon(p)| < MinPts \\ MinPts\text{-th smallest distance to } N_\varepsilon(p) & \text{otherwise} \end{cases}$$

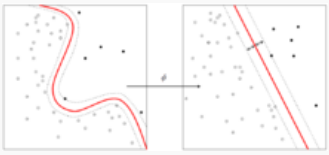
The *reachability-distance* of another point o from a point p is the distance between o and p, or the core distance of p:

$$\text{reachability-dist}_{\varepsilon,MinPts}(o,p) = \begin{cases} \text{UNDEFINED} & \text{if } |N_\varepsilon(p)| < MinPts \\ \max(\text{core-dist}_{\varepsilon,MinPts}(p), \text{dist}(p,o)) & \text{otherwise} \end{cases}$$

If p and o are nearest neighbors, this is the $\varepsilon' < \varepsilon$ we need to assume in order to have p and o belong to the same cluster.

Both the core-distance and the reachability-distance are undefined if no sufficiently dense cluster (w.r.t. ε) is available. Given a sufficiently large ε, this will never happen, but then every ε-neighborhood query will return the entire database, resulting in $O(n^2)$ runtime. Hence, the ε parameter is required to cut off the density of clusters that is no longer

Machine learning and data mining



Problems
Classification · Clustering · Regression · Anomaly detection · Association rules · Reinforcement learning · Structured prediction · Feature learning · Online learning · Semi-supervised learning · Unsupervised learning · Learning to rank · Grammar induction

Supervised learning (**classification** • **regression**)
Decision trees · Ensembles (Bagging, Boosting, Random forest) · *k*-NN · Linear regression · Naive Bayes · Neural networks · Logistic regression · Perceptron · Support vector machine (SVM) · Relevance vector machine (RVM)

Clustering
BIRCH · Hierarchical · *k*-means · Expectation-maximization (EM) · DBSCAN · **OPTICS** · Mean-shift

Dimensionality reduction
Factor analysis · CCA · ICA · LDA · NMF · PCA · t-SNE

Structured prediction
Graphical models (Bayes net, CRF, HMM)

Anomaly detection
k-NN · Local outlier factor

Neural nets
Autoencoder · Deep learning · Multilayer perceptron · RNN · Restricted Boltzmann machine · SOM · Convolutional neural network

Theory
Bias-variance dilemma · Computational learning theory · Empirical risk minimization · PAC learning · Statistical learning · VC theory

Machine learning portal
Computer science portal
Statistics portal

v · t · e

Wikipedia The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Deutsch
Français
Українська
Edit links

considered to be interesting and to speed up the algorithm this way.

The parameter ε is, strictly speaking, not necessary. It can simply be set to the maximum possible value. When a spatial index is available, however, it does play a practical role with regards to complexity. It is often claimed[*by whom?*] that OPTICS abstracts from DBSCAN by removing this parameter, at least to the extent of only having to give the maximum value.

Pseudocode [edit]

The basic approach of OPTICS is similar to DBSCAN, but instead of maintaining a set of known, but so far unprocessed cluster members, a priority queue (e.g. using an indexed heap) is used.

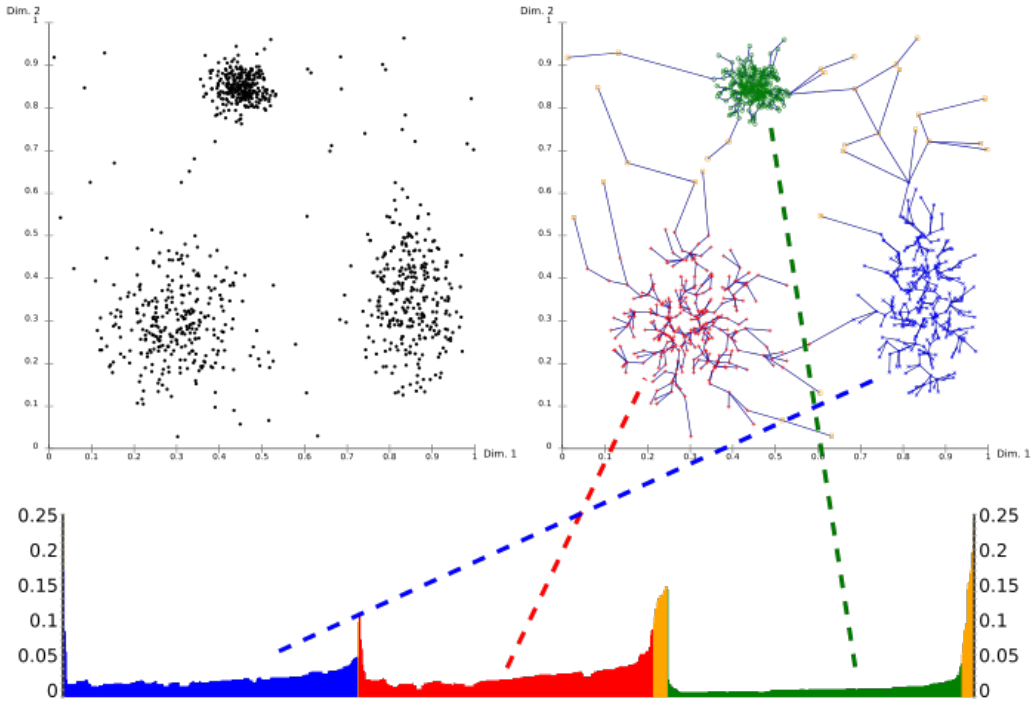
```
OPTICS(DB, eps, MinPts)
   for each point p of DB
      p.reachability-distance = UNDEFINED
   for each unprocessed point p of DB
      N = getNeighbors(p, eps)
      mark p as processed
      output p to the ordered list
      if (core-distance(p, eps, Minpts) != UNDEFINED)
         Seeds = empty priority queue
         update(N, p, Seeds, eps, Minpts)
         for each next q in Seeds
            N' = getNeighbors(q, eps)
            mark q as processed
            output q to the ordered list
            if (core-distance(q, eps, Minpts) != UNDEFINED)
               update(N', q, Seeds, eps, Minpts)
```

In update(), the priority queue Seeds is updated with the ε-neighborhood of p and q, respectively:

```
update(N, p, Seeds, eps, Minpts)
   coredist = core-distance(p, eps, MinPts)
   for each o in N
      if (o is not processed)
         new-reach-dist = max(coredist, dist(p,o))
         if (o.reachability-distance == UNDEFINED) // o is not in Seeds
             o.reachability-distance = new-reach-dist
             Seeds.insert(o, new-reach-dist)
         else               // o in Seeds, check for improvement
             if (new-reach-dist < o.reachability-distance)
                o.reachability-distance = new-reach-dist
                Seeds.move-up(o, new-reach-dist)
```

OPTICS hence outputs the points in a particular ordering, annotated with their smallest reachability distance (in the original algorithm, the core distance is also exported, but this is not required for further processing).

Extracting the clusters [edit]



Using a *reachability-plot* (a special kind of dendrogram), the hierarchical structure of the clusters can be obtained easily. It is a 2D plot, with the ordering of the points as processed by OPTICS on the x-axis and the reachability distance on the y-axis. Since points belonging to a cluster have a low reachability distance to their nearest neighbor, the clusters show up as valleys in the reachability plot. The deeper the valley, the denser the cluster.

The image above illustrates this concept. In its upper left area, a synthetic example data set is shown. The upper right part visualizes the spanning tree produced by OPTICS, and the lower part shows the reachability plot as computed by OPTICS. Colors in this plot are labels, and not computed by the algorithm; but it is well visible how the valleys in the plot correspond to the clusters in above data set. The yellow points in this image are considered noise, and no valley is found in their reachability plot. They will usually not be assigned to clusters except the omnipresent "all data" cluster in a hierarchical result.

Extracting clusters from this plot can be done manually by selecting a range on the x-axis after visual inspection, by selecting a threshold on the y-axis (the result will then be similar to a DBSCAN clustering result with the same ε and minPts parameters; here a value of 0.1 may yield good results), or by different algorithms that try to detect the valleys by steepness, knee detection, or local maxima. Clusterings obtained this way usually are hierarchical, and cannot be achieved by a single DBSCAN run.

Complexity [edit]

Like DBSCAN, OPTICS processes each point once, and performs one ε-neighborhood query during this processing. Given a spatial index that grants a neighborhood query in $O(\log n)$ runtime, an overall runtime of $O(n \cdot \log n)$ is obtained. The authors of the original OPTICS paper report an actual constant slowdown factor of 1.6 compared to DBSCAN. Note that the value of ε might heavily influence the cost of the algorithm, since a value too large might raise the cost of a neighborhood query to linear complexity.

In particular, choosing $\varepsilon > \max_{x,y} d(x,y)$ (larger than the maximum distance in the data set) is possible, but will obviously lead to quadratic complexity, since every neighborhood query will return the full data set. Even when no spatial index is available, this comes at additional cost in managing the heap. Therefore, ε should be chosen appropriately for the data set.

Extensions [edit]

OPTICS-OF[3] is an outlier detection algorithm based on OPTICS. The main use is the extraction of outliers from an existing run of OPTICS at low cost compared to using a different outlier detection method.

DeLi-Clu,[4] Density-Link-Clustering combines ideas from single-linkage clustering and OPTICS, eliminating the ε parameter and offering performance improvements over OPTICS.

HiSC[5] is a hierarchical subspace clustering (axis-parallel) method based on OPTICS.

HiCO[6] is a hierarchical correlation clustering algorithm based on OPTICS.

DiSH[7] is an improvement over HiSC that can find more complex hierarchies.

FOPTICS[8] is a faster implementation using random projections.

Availability [edit]

Implementations of OPTICS, OPTICS-OF, DeLi-Clu, HiSC, HiCO and DiSH are available in the ELKI data mining framework (with index acceleration). An incomplete and slow implementation can be found in the Weka extensions. The Francis Crick Institute provides a C reimplementation of OPTICS without index support.

References [edit]

1. ^ Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, Jörg Sander (1999). *OPTICS: Ordering Points To Identify the Clustering Structure*. ACM SIGMOD international conference on Management of data. ACM Press. pp. 49–60.
2. ^ Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu (1996). Evangelos Simoudis, Jiawei Han, Usama M. Fayyad, eds. *A density-based algorithm for discovering clusters in large spatial databases with noise*. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press. pp. 226–231. ISBN 1-57735-004-9.
3. ^ Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng and Jörg Sander (1999). "OPTICS-OF: Identifying Local Outliers". *Principles of Data Mining and Knowledge Discovery*. Springer-Verlag. pp. 262–270. doi:10.1007/b72280. ISBN 978-3-540-66490-1.
4. ^ Achtert, E.; Böhm, C.; Kröger, P. (2006). "DeLi-Clu: Boosting Robustness, Completeness, Usability, and Efficiency of Hierarchical Clustering by a Closest Pair Ranking". *LNCS: Advances in Knowledge Discovery and Data Mining*. Lecture Notes in Computer Science **3918**: 119–128. doi:10.1007/11731139_16. ISBN 978-3-540-33206-0.
5. ^ Achtert, E.; Böhm, C.; Kriegel, H. P.; Kröger, P.; Müller-Gorman, I.; Zimek, A. (2006). "Finding Hierarchies of Subspace Clusters". *LNCS: Knowledge Discovery in Databases: PKDD 2006*. Lecture Notes in Computer Science **4213**: 446–453. doi:10.1007/11871637_42. ISBN 978-3-540-45374-1.
6. ^ Achtert, E.; Böhm, C.; Kröger, P.; Zimek, A. (2006). "Mining Hierarchies of Correlation Clusters". *Proc. 18th International Conference on Scientific and Statistical Database Management (SSDBM)*: 119–128. doi:10.1109/SSDBM.2006.35. ISBN 0-7695-2590-3.
7. ^ Achtert, E.; Böhm, C.; Kriegel, H. P.; Kröger, P.; Müller-Gorman, I.; Zimek, A. (2007). "Detection and Visualization of Subspace Cluster Hierarchies". *LNCS: Advances in Databases: Concepts, Systems and Applications*. Lecture Notes in Computer Science **4443**: 152–163. doi:10.1007/978-3-540-71703-4_15. ISBN 978-3-540-71702-7.
8. ^ Schneider, Johannes; Vlachos, Michail (2013). "Fast parameterless density-based clustering via random projections". *22nd ACM International Conference on Information and Knowledge Management(CIKM)* (ACM).

Categories: Data clustering algorithms