Conclusions

What this handout is about

This handout will explain the functions of conclusions, offer strategies for writing effective ones, help you evaluate your drafted conclusions, and suggest conclusion strategies to avoid.

About conclusions

Introductions and conclusions can be the most difficult parts of papers to write. While the body is often easier to write, it needs a frame around it. An introduction and conclusion frame your thoughts and bridge your ideas for the reader.

Just as your introduction acts as a bridge that transports your readers from their own lives into the "place" of your analysis, your conclusion can provide a bridge to help your readers make the transition back to their daily lives. Such a conclusion will help them see why all your analysis and information should matter to them after they put the paper down.

Your conclusion is your chance to have the last word on the subject. The conclusion allows you to have the final say on the issues you have raised in your paper, to synthesize your thoughts, to demonstrate the importance of your ideas, and to propel your reader to a new view of the subject. It is also your opportunity to make a good final impression and to end on a positive note.

Your conclusion can go beyond the confines of the assignment. The conclusion pushes beyond the boundaries of the prompt and allows you to consider broader issues, make new connections, and elaborate on the significance of your findings.

Your conclusion should make your readers glad they read your paper. Your conclusion gives your reader something to take away that will help them see things differently or appreciate your topic in personally relevant ways. It can suggest broader implications that will not only interest your reader, but also enrich your reader's life in some way. It is your gift to the reader.

Strategies for writing an effective conclusion

One or more of the following strategies may help you write an effective conclusion.

- Play the "So What" Game. If you're stuck and feel like your conclusion isn't saying anything new or interesting, ask a friend to read it with you. Whenever you make a statement from your conclusion, ask the friend to say, "So what?" or "Why should anybody care?" Then ponder that question and answer it. Here's how it might go:

  You: *Basically, I'm just saying that education was important to Douglass.*

  Friend: *So what?*

  You: *Well, it was important because it was a key to him feeling like a free and equal citizen.*

  Friend: *Why should anybody care?*

  You: *That's important because plantation owners tried to keep slaves from being educated so that they could maintain control. When Douglass obtained an education, he undermined that control personally.*

  You can also use this strategy on your own, asking yourself "So What?" as you develop your ideas or your draft.

- Return to the theme or themes in the introduction. This strategy brings the reader full circle. For example, if you begin by describing a scenario, you can end with the same scenario as proof that your essay is helpful in creating a new understanding. You may also refer to the introductory paragraph by using key words or parallel concepts and images that you also used in the introduction.
- Synthesize, don't summarize: Include a brief summary of the paper's main points, but don't simply repeat things that were in your paper. Instead, show your reader how the points you made and the support and examples you used fit together. Pull it all together.
- Include a provocative insight or quotation from the research or reading you did for your paper.
- Propose a course of action, a solution to an issue, or questions for further study. This can redirect your reader's thought process and help her to apply your info and ideas to her own life or to see the broader implications.

Future work

- Point to Broader Implications
- What did we learn from the work

Abstract: In this paper, we introduce a new clustering algorithm for spatial-temporal data based on DBSCAN (Density-Based Spatial Clustering Applications with Noise) method. Our algorithm, ST-DBSCAN, improves DBSCAN method in three important directions: clustering spatial-temporal data based on its non-spatial, spatial and temporal attributes, detecting noise points using different density factor for each cluster and identifying adjacent clusters by comparing the average value of a cluster with new coming value. Moreover, in this paper, we present a spatial data warehouse system, which is designed for the purpose of storing and clustering a wide range of spatial-temporal data. Experiments are conducted to demonstrate the applicability of our algorithm to real world problems.

Vague

Conclusion: We have presented an algorithm for clustering spatial-temporal data which improves the DBSCAN method and shows potential applicability in solving real world problems. The paper's ~~proposed~~ makes ~~algorithm consists of~~ three main contributions: the two distance metrics for spatial values and non-spatial values, the density factor for noise detection and the average comparison for identifying adjacent clusters. The experimental results suggest that ST-DBSCAN is robust for clustering spatial-temporal data. The processing time is significantly improved by our data warehouse system. We utilizes the R-Tree indexing method to handle spatial-temporal information, and we also adopts some filters to reduce the search space for spatial data mining algorithm.

Short!

A data warehouse was designed

A 'better' version of the abstract above

Arthur Dubbas

Conclusion:

While DBSCAN is useful in many contexts, it has some problems in practice because it was not designed specifically to handle spatial-temporal data, and it has difficulty recognizing clusters when the data has regions with variable density. In this paper we introduced ST-DBSCAN, an improvement on DBSCAN that fixes many of DBSCAN's problems while maintaining the same order of algorithmic time complexity, O(n*logn).

how was it made

?!

Spatial data and temporal data are different from other data types because they come with a set of explicit relationships in space and time. Clustering algorithms that are made for spatial-temporal data, like ST-DBSCAN, give much more accurate information than algorithms that were not designed with these relationships in mind.

First, unlike DBSCAN, ST-DBSCAN is designed to cluster points based upon non-spatial, spatial, and temporal data. Second, ST-DBSCAN is able to handle clusters with varying densities because it assigns each cluster a 'density factor' which allows the distance swept out from a point to vary with the density of the region. Finally, ST-DBSCAN prevents clusters from artificially expanding to include large numbers of noise points by comparing each new point that would be assigned to a cluster against the average values in the cluster. [Together, these advantages allow ST-DBSCAN to recognize patterns that normal DBSCAN may miss, and to more easily thread the needle between clustering a diffuse cluster together correctly and avoiding including noise points in to a densely packed cluster.] 1

Good! Some form of integra!

We also developed a spatial data warehouse system that works well with ST-DBSCAN.

What is the broader impact of the Statement

Finally, we demonstrated ST-DBSCAN's effectiveness on real world data.

[In future work, we plan to run ST-DBSCAN on data sets that have explicit classes to see how well our clusters discover actual already recognized underlying similarities in data. We also plan to compare its results against multiple other spatial-temporal clustering algorithms to see how well ST-DBSCAN performs relative to its direct peers.] 2

Boundaries / Broader Issues Implied

Synthesis; integrate things from all parts of the paper (e.g. also the introduction)

What can the reader take home

Kunal

Conclusion

Clustering is one of the important methods for knowledge discovery and data mining applications. Spatial-temporal data is being generated in large amounts and needs to be analyzed. The paper introduces a new density-based clustering algorithm for clustering spatial-temporal data. The existing density-based algorithms are not designed to cluster spatial, non-spatial and temporal data. Moreover, they cannot discover clusters of different densities or adjacent clusters. The proposed algorithm ~~to able~~ to overcome these difficulties by introducing similarity measures for spatial ~~and~~ non-temporal data (Eps1 and Eps2), a density factor which signifies the density of each cluster and a threshold value for the difference between average value of a ~~cluster and an object~~ value. The algorithm works in a similar manner as the DBSCAN algorithm and also has a runtime complexity very similar to the DBSCAN algorithm.

focus more on capabilities

hard to understand

?

A spatial-temporal warehouse is used to demonstrate the applicability of the algorithm to real world problems. First application was discovering regions with similar sea temperature values. The algorithm was able to generate clusters with similar characteristics. It discovered regions of the sea nearby colder regions in the north (geographically) as clusters and regions of the seas with hotter temperatures in the south (geographically) closer to the equator as clusters. The second application was finding regions with similar sea surface height values. It discovered clusters in the three seas with similar characteristics. The third application was discovering regions with similar wave height values. The algorithm was able to cluster regions which had significantly similar wave heights; some of them with heights as low as 0.5 meters and some of them with heights as high as 3.6 meters.

Does the warehouse demonstrate the applicability

Summarize with respect to change

Seems to be to show that the algorithm is useful!

good: future work

An algorithm like ST-DBSCAN designed for clustering spatial-temporal data can be used in applications such as geographical information systems, medical imaging, and weather forecasting. Some examples of applications can be detecting changes in vegetation vigor, town planning according to commuting patterns, etc. A similar application would be in trajectory clustering where you cluster data based on the trajectories of objects in the spatial-temporal domain. Future research directions could be in developing spatial-temporal clustering algorithms that improve on other types of clustering methods, i.e. i) partition algorithms (k-means, k-medoids), ii) hierarchical algorithms (CURE, BIRCH), iii) grid-based algorithms (STING, WaveCluster) and iv) model-based algorithms (COBWEB). Other direction of research would be outlier/anomaly detection in large databases in spatial-temporal domain.

algorithm already does this

?!: Looks more like future work

Nikolaos Sarafianos – 1392654

Professor: Dr. Eick

Advanced Machine Learning – Homework 1

Th. 10.29.2015

ST-DBSCAN: An algorithm for clustering spatial-temporal data

Abstract:

Spatio-temporal clustering is a process of grouping objects based on their spatial and temporal similarity. In this paper, we present a new density-based clustering algorithm ST-DBSCAN which is based on DBSCAN. We selected DBSCAN algorithm due to (i) its ability in discovering clusters with arbitrary shape; (ii) the fact that it does not require predetermination of the number of clusters; and (iii) its ability to process large databases. The proposed algorithm improves DBSCAN without changing its runtime complexity in three ways: (i) can cluster spatial-temporal data according to its non-spatial, spatial and temporal attributes; (ii) can detect some noise points when clusters of different densities exist by assigning a density factor to each cluster; and (iii) solves the problem of completely different values in the object borders of 2 sides in a cluster. We applied ST-DBSCAN to a spatial data warehouse system we designed for the purpose of this paper and present our results in 3 data mining applications.

Conclusion: Short!

In this paper we introduced ST-DBSCAN, a new density-based clustering algorithm based on DBSCAN which exploits its key characteristics and at the same time improves its limitations. ST-DBSCAN is capable of clustering spatial-temporal data according to its non-spatial, spatial and temporal attributes and by comparing the average value of each cluster with new coming value, it tackles the problem that appears when the values in the object borders of 2 sides of a cluster are different. We also introduced a density factor which is assigned to each cluster in order to address the problem of noise points in cases where clusters have different densities. Finally, we presented 3 data mining applications of our approach by designing a spatial-temporal data warehouse which contains geographical information about different seas and described the process of KDD in each step.

??

not clear

What are values here?

Reads more an abstract

Assigned Paper: ST-DBSCAN: An Algorithm for Clustering Spatial-temporal data

Student: Pham Dinh Nguyen

Abstract

This work proposes an algorithm to cluster spatial temporal data. The algorithm is ST-DBSCAN, based on the well-known density-based DBSCAN algorithm. DBSCAN strength is scalability and ability to discover cluster of arbitrary shapes. We further extend DBSCAN with three important contribution: clustering spatial temporal data according to its non-spatial, spatial and temporal attributes; efficiently detecting noise points in the presence of different densities; stabilizing border points detection for adjacent clusters. We also present a data warehouse system, which provides storage, management, clustering analysis, and visualization for a wide range of spatial temporal data. We use that system to demonstrate our algorithm and discuss the clustering results.

Conclusion English

at least diffent from abstract

In this study, we introduce ST-DBSCAN with three major improvements to DBSCAN ~~and~~ to mitigate its current limitations. ~~The~~ Its main extension is the ability to cluster spatial temporal data. The second extension is the introduction of density factor, allowing the detection of clusters with different densities, which is useful ~~in analyze~~ when analyzing real data sets. The third improvement is the use of the mean value of the cluster to robustly determine border points, which is critical for adjacent clusters. Why is it critical?

We demonstrate our algorithm with real weather sensing data set, using our own data warehouse. The results are visualized and presented in a user-friendly interface, showing some interesting findings. ⟹ Broader Impact?!? Value of what was created? Value? ?!?

Future work

As an extension over DBSCAN, our algorithm requires more preset thresholds, as shown in the pseudo code. Though we only mention one heuristic method to choose these parameters, we believe there can be others heuristics, considering the spatial temporal relation, would be worth

Our implementation has the same performance with DBSCAN, which is O(n*log(n)), which is among ~~the best runtime~~ in clustering methods. We further improve the performance by apply R-Tree index in our database, and ~~were~~ to be able to process large real data sets. Further improvement might be in the direction of parallelizing the warehouse system. Clarify

K-means is O(Kn)

Sparse – Use more words to get your message across.