ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 129-134

Distributed Clustering and Local Regression for Knowledge Discovery in Multiple Spatial Databases

Aleksandar Lazarevic, Dragoljub Pokrajac, Zoran Obradovic

School of Electrical Engineering and Computer Science, Washington
State University, Pullman, WA 99164-2752, USA
{alazarev, dpokraja, zoran}@eecs.wsu.edu

Abstract. *Many large-scale spatial data analysis problems involve an investigation of relationships in heterogeneous databases. In such situations, instead of making predictions uniformly across entire spatial data sets, in a previous study we used clustering for identifying similar spatial regions and then constructed local regression models describing the relationship between data characteristics and the target value inside each cluster. This approach requires all the data to be resident on a central machine, and it is not applicable when a large volume of spatial data is distributed at multiple sites. Here, a novel distributed method for learning from heterogeneous spatial databases is proposed. Similar regions in multiple databases are identified by independently applying a spatial clustering algorithm on all sites, followed by transferring convex hulls corresponding to identified clusters and their integration. For each discovered region, the local regression models are built and transferred among data sites. The proposed method is shown to be computationally efficient and fairly accurate when compared to an approach where all the data are available at a central location.*

Introduction

The number and the size of spatial databases are rapidly growing in various GIS applications ranging from remote sensing and satellite telemetry systems, to computer cartography and environmental planning. Many large-scale spatial data analysis problems also involve an investigation of relationships among attributes in heterogeneous data sets. Therefore, instead of applying global recommendation models across entire spatial data sets, they are varied to better match site-specific needs thus improving prediction capabilities [1]. Our recently proposed approach towards such a modeling is to define spatial regions having similar characteristics, and to build local regression models on them describing the relationship between the spatial data characteristics and the target attribute [2].

However, spatial data is often inherently distributed at multiple sites and cannot be localized on a single machine for a variety of practical reasons including physically dispersed large data sets over many different geographic locations, security services and competitive reasons. In such situations, the proposed approach of building local regressors [2] can not be applied, since the data needed for clustering can not be centralized on a single site. Therefore, there is a need to improve this method to learn from large spatial databases located at multiple data sites.

A new viable approach for distributed learning of locally adapted models is explored in this paper. Given a number of distributed, spatially dispersed data sets, we first define more homogenous spatial regions in each data set using a distributed clustering algorithm. The next step is to build local regression models and transfer them among the sites. Our experimental results showed that this method is computationally effective and fairly accurate when compared to an approach where all data are localized at a central machine.

Methodology

Partitioning spatial data sets into regions having similar attribute values should result in regions of similar target value. Therefore, using the relevant features, a spatial clustering algorithm is used to partition each spatial data set independently into “similar” regions. A clustering algorithm is applied in an unsupervised manner (ignoring the target attribute value). As a result, a number of partitions (clusters) on each spatial data set is obtained. Assuming similar data distributions of the observed data sets, this number of clusters on each data set is usually the same (Figure 1). If this is not the case, by choosing the appropriate clustering parameter values the discovery of an identical number of clusters on each data set can be easily enforced.

The next step is to match the clusters among the distributed sites, i.e. which cluster from one data set is the most similar to which cluster in another spatial data set. This is followed by building the local regression models on identified clusters at sites with known target attribute values. Finally, learned models are transferred to the remaining sites where they are integrated and applied to estimate unknown target values at the appropriate clusters.

2.1. Learning at a single site

Although the proposed method can be applied to an arbitrary number of spatial data sets, for the sake of simplicity assume first that we predict on the set D_2 by using local regression models built on the set D_1. Each of *k* clusters $C_{1,i}$, $i = 1,...k$, identified at D_1 ($k = 5$ at Figure 1), is used to construct a corresppnding local regression model M_i.

To apply local models trained on D_1 subsets to unseen data set D_2 we construct a convex hull for each cluster on the data set D_1, and transfer all convex hulls to a site containing unseen data set D_2 (Figure 1). Using the convex hulls of the clusters from D_1 (shown with solid lines in Figure 1), we identify the correspondence between the clusters from two spatial data sets. This is determined by identifying the best matches between the clusters $C_{1,i}$ (from the set D_1) and the clusters $C_{2,i}$ (from the set D_2). For example, the convex hull $H_{1,4}$ at Figure 1 covers both the clusters $C_{2,5}$ and $C_{2,4}$, but it covers $C_{2,5}$ in much larger fraction than it covers $C_{2,4}$. Therefore, we concluded that the cluster $C_{1,4}$ matches the cluster $C_{2,5}$, and the local regression model M_4 built on the cluster $C_{1,4}$ is applied to the cluster $C_{2,5}$.

However, there are also situations where the exact matching can not be determined, since there are significant overlapping regions between the clusters from different data sets (e.g. the convex hull $H_{1,1}$ covers both the clusters $C_{2,2}$ and $C_{2,3}$ on Figure 1, and there is an overlapping region O_1). To improve the prediction, the combination of the local regression models built on neighboring clusters is used on overlapping regions. For example, the prediction for the region O_1 at Figure 1 is made using the simple averaging of local prediction models learned on the clusters $C_{1,1}$ and $C_{1,5}$.

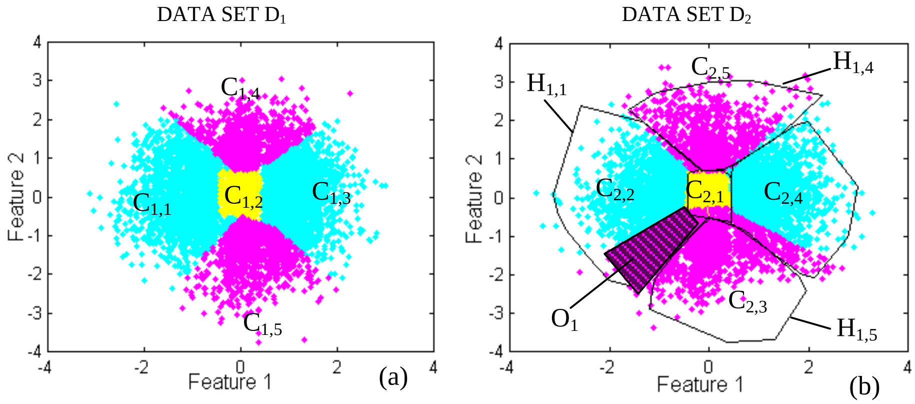

Figure 1. Clusters in the feature space for two spatial data sets: D_1 and D_2 and convex hulls ($H_{1,i}$) from data set D_1 (a) transferred to the data set D_2 (b).

However, there are also situations where the exact matching can not be determined, since there are significant overlapping regions between the clusters from different data sets (e.g. the convex hull $H_{1,1}$ covers both the clusters $C_{2,2}$ and $C_{2,3}$ on Figure 1, so there is an overlapping region O_1). To improve the prediction, averaging of the local regression models built on neighboring clusters is used on overlapping regions. For example, the prediction for the region O_1 at Figure 1 is made using the simple averaging of local prediction models learned on the clusters $C_{1,1}$ and $C_{1,5}$. In this way we hope to achieve better prediction accuracy than local predictors built on entire clusters.

2.2. Learning from multiple data sites

When data from more physically distributed sites are available for modeling, the prediction can be further improved by integrating learned models from several data sites. Without loss of generality, assume there are 3 dispersed data sites, where the prediction is made on the third data set (D_3) using the local prediction models from the first two data sets D_1 and D_2. The key idea is the same as in the two data sets scenario, except more overlapping is likely to occur in this scenario. To simplify the presentation, we will discuss the algorithm only for the matching clusters $C_{1,1}$, $C_{2,2}$ and $C_{3,2}$ from the data sets D_1, D_2 and D_3 respectively (Figure 2).

The intersection of $H_{1,1}$, $H_{2,2}$ and $C_{3,2}$ (region C) represents the portion of the cluster $C_{3,2}$, where clusters from all three fields are matching. Therefore, the prediction on this region is made by averaging the models built on the clusters $C_{1,1}$ and $C_{2,2}$, whose contours are represented in Figure 2 by convex hulls $H_{1,1}$ and $H_{2,2}$, respectively. Making the predictions on the overlapping portions O_i, i = 1,2,3 is similar to learning at a single site. For example the prediction on the overlapping portion O_1 is made by averaging of the models learned on the clusters $C_{1,1}$, $C_{2,2}$ and $C_{2,3}$.

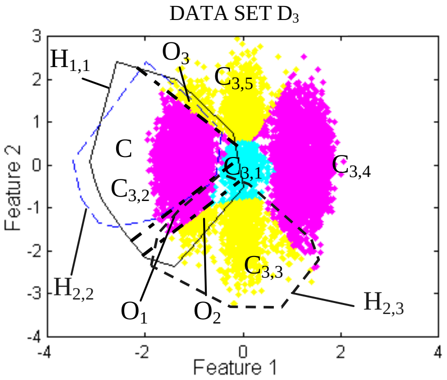

Figure 2. Transferring the convex hulls from two sites with spatial data sets to a third site

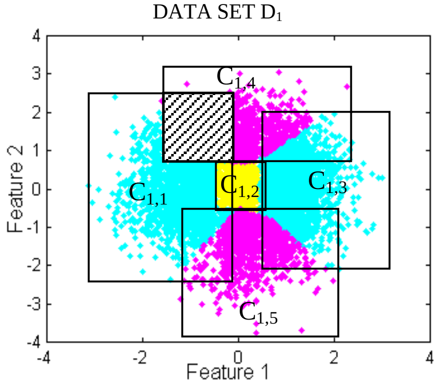

Figure 3. The alternative representation of the clusters with MBRs

2.3. The comparison to minimal bounding rectangle representation

An alternative method of representing the clusters, popular in database community, is to construct a minimal bounding rectangle (MBR) for each cluster. The apparent advantages of this approach are limiting the data transfer further, since the MBR can be represented by less data points than convex hulls, and reducing the computational complexity from $\Theta(n \cdot log\ n)$ for computing a convex hull of n points to $\Theta(n)$ for computing a corresponding MBR. However, this approach results in large overlapping of neighboring clusters (see shadowed part on Figure 3). Therefore, using a convex hull based algorithm leads to a much better cluster representation for the price of slightly increasing the computational time and the data transfer rate.

Experimental Results

Our experiments were performed using artificial data sets generated using our spatial data simulator [4] to mix 5 homogeneous data distributions, each having different relevant attributes for generation of the target attribute. Each data set had 6561 patterns with 5 relevant attributes, where the degree of relevance was different for each distribution. Spatial clustering is performed using a density based algorithm DBSCAN [5], which was previously used in our centralized spatial regression modeling.

As local regression models, we trained 2-layered feedforward neural network models with 5, 10 and 15 hidden neurons. We used Levenberg-Marquardt [3] learning algorithm and repeated experiments starting from 3 random initializations of network parameters. For each of these models, the prediction accuracy was measured using the coefficient of determination defined as $R^2 = 1 - MSE/\sigma^2$, where σ is a standard deviation of the target attribute. R^2 value is a measure of the explained variability of the target variable, where 1 corresponds to a perfect prediction, and 0 to a trivial mean predictor.

Method	R^2 ± std
Global model	0.73±0.01
Matching clusters	0.82±0.02
Matching clusters + averaging for overlapping regions	0.87±0.03
Centralized clustering (upper bound)	0.87±0.02

Table1. Models built on set D_1 applied on D_2

Method	R^2 value ± std	
	combine models from	
	single site	all sites
Global models	0.75±0.02	0.77±0.02
Matching clusters	0.89±0.02	0.90±0.02
Matching clusters + averaging for overlapping regions	0.90±0.02	0.92±0.03
Centralized clustering (upper bound)	0.90±0.01	0.92±0.02

Table 2. Models built on sets D_1 and D_2 applied to D_3

When constructing regressors using spatial data from a single site and testing on spatial data from another site, the prediction accuracies averaged over 9 experiments are given in the Table 1. The accuracy of local specific regression models significantly outperformed the global model trained on all D_1 data. By incorporating the model combinations on significant overlapping regions between clusters, the prediction capability was improved. This indicated that indeed confidence of the prediction in the overlapping parts can be increased by averaging appropriate local predictors. In summary, for this data set, the proposed distributed method can successfully approach the upper bound of centralized technique, where two spatial data sets are merged together at a single site and when the clustering is applied to the merged data set.

The prediction changes depending on the noise level, the number and the type of noisy features (features used for clustering and modeling or for modeling only). We have experimented with adding different levels of Gaussian noise to clustering and modeling features (5%, 10% and 15%) for the total number of noisy features ranging from 1 to 5. (Figure 4).

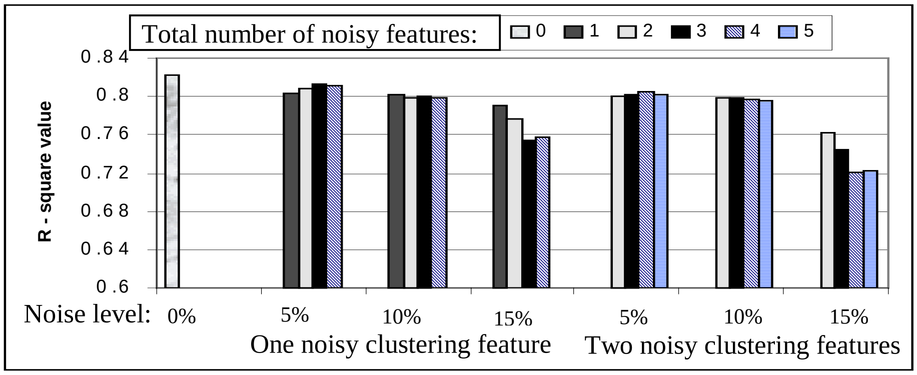

Figure 4. The influence of different noise levels on the prediction accuracy. We added none, 1, 2 and 3 noisy modeling features to the 1 or 2 noisy clustering features. We have experimented with 5%, 10% and 15% of noise level. We used matching clusters.

Figure 4 shows that when a small noise is present in features (5%, 10%), even if some of them are clustering features, the method is fairly robust. However, by increasing the noise level (15%), the prediction accuracy starts to decrease significantly.

Finally, when models from 2 distributed data sites are combined to make prediction on the third spatial data set, the prediction accuracy was improved more than when considering only the models from a single site (Table 2). The influence of the noise is similar in this case, and the experimental results are omitted for lack of space.

Conclusions

Experiments on two and three simulated heterogeneous spatial data sets indicate that the proposed method for learning local site-specific models in a distributed environment can result in significantly better predictions as compared to using a global model built on the entire data set. When comparing the proposed approach to a centralized method (all data are available at the single data site), we observe no significant difference in the prediction accuracy achieved on the unseen spatial data sets. The communication overhead of data exchange among the multiple data sites is small, since only the convex hulls and the models built on the clusters are transferred. Furthermore, the suggested algorithm is very robust to small amounts of noise in the input features.

Although the performed experiments provide evidence that the proposed approach is suitable for distributed learning in spatial databases, further work is needed to optimize methods for combining models in larger distributed systems. We are currently extending the method to a distributed scenario with different sets of known features at various databases.

References

1. Hergert, G., Pan, W., Huggins, D., Grove, J., Peck, T., "Adequacy of Current Fertilizer Recommendation for Site-Specific Management", In Pierce F., "The state of Site-Specific Management for Agriculture," *American Society for Agronomy*, *Crop Science Society of America*, *Soil Science Society of America*, chapter 13, pp. 283-300, 1997.
2. Lazarevic, A., Xu, X., Fiez, T. and Obradovic, Z., “Clustering-Regression-Ordering Steps for Knowledge Discovery in Spatial Databases,” *Proc. IEEE/INNS Int’l Conf. on Neural Neural Networks*, Washington, D.C., July 1999, ISBN 0-7803-5532-0, No. 345, Session 8.1B.
3. Hagan, M., Menhaj, M.B.: Training feedforward networks with the Marquardt algorithm. IEEE Transactions on Neural Networks Vol. 5, pp. 989-993, 1994.
4. Pokrajac, D., Fiez, T. and Obradovic, Z.: A Spatial Data Simulator for Agriculture Knowledge Discovery Applications, in review.
5. Sander J., Ester M., Kriegel H.-P., Xu X.: “Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications,” *Data Mining and Knowledge Discovery, An International Journal*, Kluwer Academic Publishers, Vol. 2, No. 2, pp. 169-194, 1998.