A New Geometric Approach of Adaptive Neighborhood Selection for Classification

Editors: List of editors' names

Abstract

The k-nearest neighbor (k-NN) is a widely adopted technique for nonparametric classification. However, the specification of the number of neighbors, k, often presents a challenge and highlights relevant constraints. Many desirable characteristics of a classifier - including the robustness to noise, smoothness of decision boundaries, bias-variance tradeoff, and management of class imbalance - are directly impacted by this parameter. In the present work, we describe an adaptive k-nearest-neighbors method that locally defines the neighborhood size by investigating the curvature of the sample. The rationale is that points with high curvature may have smaller neighbors (locally, the tangent space is a loose approximation) and points with low curvature may have larger neighborhoods (locally, the tangent space approximates the underlying data shape well). The results on several real-world data sets indicate that the new method outperforms the well-established k-NN approach.

Keywords: Curvature, k-nearest neighbors, Shape operator, Supervised classification.

1. Introduction

The nonparametric technique for pattern classification known as the k-nearest neighbor classifier (k-NN) is renowned for its ease of use, adaptability, and intuitive approach (Cover and Hart, 1967; Nielsen, 2016). The neighborhood size is controlled by the parameter k, which drives the behavior of the k-NN classifier. When generating predictions for a new data point, this parameter indicates the number of nearest neighbors considered (Jodas et al., 2022). A more flexible method with decision bounds that closely match the training data is provided by a lower k value, which may be capable of capturing complex patterns and local changes. Nevertheless, due to the fact that it overly depends on the closest neighbors for classification, lower k values should increase the vulnerability to problems related noise and outliers (Uddin et al., 2022). On the other hand, larger k values produce a more generic method that is less influenced by single data points while considering a smoother decision boundary.

In the present work, we describe a geometric approach for adaptive neighborhood selection in the k-nearest neighbor classifier, which automatically modify the number of neighbors for each sample. Through the inspection of the local curvature, the adaptive approach of the curvature-based K-NN classifier defines the neighborhood size k at each vertex of the k-nearest neighbors graph (k-NNG). The tangent plane is frequently tightly tuned to a manifold in the case of points with smaller curvature values, allowing the definition of larger neighborhoods. However, the tangent plane is loose considering points of high curvature, reducing the size of the neighborhood. Our findings over several cases indicate a superior performance of the proposed curvature-based method compared to the regular and widely adopted k-NN method.

2. Local shape operator and curvatures

A challenging problem is to appropriately estimating the curvature at each sample of the data set. We propose an algorithm to estimate the local shape operator based on discrete approximations to the local metric and curvature tensors - i.e., generalizations of the first and second fundamental forms of a surface (Levada et al., 2024). This aims to approximate the metric tensor as the inverse of the local covariance matrix as well as the curvature tensor as the local Hessian matrix, based on the definitions of the Hessian Eigenmaps algorithm (Donoho and Grimes, 2003).

Algorithm 1: The shape operator-based curvatures

```
1. $A \leftarrow k\text{NN-graph}(X, k)$    // Builds the kNN-graph from data matrix X
2. For $k = 1$ to $n$    // Scan each sample
    (a) $neighbors \leftarrow N(\boldsymbol{x}_i)$    // Neighborhood of sample $\boldsymbol{x}_i$
    (b) $\Sigma_i \leftarrow \text{cov-matrix(neighbors)}$    // Local covariance matrix
    (c) $U \leftarrow eigenvectors(\Sigma_i)$    // Eigenvectors = columns of U
    (d) Compute the matrix $X_i$ with $1 + m + m(m+1)/2$ columns
    (e) Compute the matrix $H_i$: the last $m(m+1)/2$ columns of $X_i$
    (f) $\mathcal{H}_i \leftarrow \hat{H}_i \hat{H}_i^T$    // Second fundamental form
    (g) $\mathcal{S}_i \leftarrow -\mathcal{H}_i \Sigma_i$    // Shape operator
    (h) $K_i \leftarrow det(\mathcal{S}_i)$    // Curvature at point $\boldsymbol{x}_i$
3. Return $K$    // The vector of local curvatures
```

3. Curvature-based approach for adaptive k-NN

Subsequently to the computation of the curvature in each sample, a total of ten distinct scores (ranging from zero to nine) are assigned to the curvatures. The edges of the k-NNG are pruned in order to perform the adaptive neighborhood adjustment, given the scores determined by the local curvatures. For example, assuming that $k = 11$ and sample $\boldsymbol{x}_i$ has a curvature score of $c_i = 4$, the neighborhood would consist of only seven neighbors. The edges connecting sample $\boldsymbol{x}_i$ with its four farthest neighbors would be removed. The sample $\boldsymbol{x}_i$ would still be linked to its closest neighbor when $k < c_i$. During the testing phase, the new sample $\boldsymbol{z}_i$ is included and classified, determining its local curvature while connecting it to its k-nearest neighbors. Lastly, the new point's curvature is added to the vector of curvatures, generating its score.

Table 1: Sample size, number of features, and classes of the selected openML data sets of the first round of experiments.

#	Data set	# samples	# features	# classes
1	`vowel`	990	13	11
2	`tecator`	240	124	2
3	`sonar`	208	60	2
4	`ionosphere`	351	34	2
5	`user-knowledge`	403	5	5
6	`parkinsons`	195	22	2
7	`breast-tissue`	106	9	4
8	`Smartphone-Based_Recognition`	180	66	6
9	`mfeat-fourier` (25%)	500	76	10
10	`letter` (10%)	2000	16	26
11	`satimage` (25%)	1607	36	6
12	`pendigits` (25%)	2748	16	10
13	`texture` (25%)	1375	40	11
14	`digits` (25%)	449	64	10
15	`Olivetti_Faces (10 LDA features)`	400	10	40

4. Preliminary results

We selected 15 data sets from distinct domains, with a varying number of features. The proposed adaptive K-NN is compared with the standard k-NN classifier considering their balanced accuracy, Kappa coefficient, Jaccard index, and F1-score.

A holdout strategy is implemented to separate the samples into training and test data sets. With 5% increments, the training partition may range from 10% to 90% of the total samples. This means that there are a total of 17 potential divisions during the training and testing phases. The rationale is to test the behavior of the methods while exploring small, medium and large training sets. The preliminary results are reported in Table 2. The proposed curvature-based k-NN method outperforms the regular k-NN for all data sets.

5. Conclusion

In this work, we propose a curvature-based k-NN classification algorithm. The new method is devised to increase classification accuracy by locally adjusting the neighborhood size while exploring the intrinsic curvature information of the data set. Our experimental findings provide insights into the efficiency and adaptability of the proposed approach. Future works may include a weighted version of the curvature-based k-NN.

References

T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967. doi: 10.1109/TIT.1967.1053964.

Table 2: Median of measures after 17 executions adopting the holdout strategy, with training data sets of different sizes: from 10% to 90% of with increments of 5%

Data set	*k*-NN				Proposed method			
	Bal. Acc.	Kappa	Jaccard	F1	Bal. Acc.	Kappa	Jaccard	F1
1	0.5314	0.4679	0.3430	0.4978	**0.8860**	**0.8732**	**0.7999**	**0.8828**
2	0.7779	0.5698	0.6479	0.7843	**0.8308**	**0.6689**	**0.7142**	**0.8333**
3	0.7227	0.4547	0.5712	0.7240	**0.8285**	**0.6599**	**0.7121**	**0.8315**
4	0.7326	0.5146	0.6568	0.7839	**0.8205**	**0.6824**	**0.7484**	**0.8534**
5	0.5371	0.6021	0.5498	0.6791	**0.5872**	**0.6661**	**0.6085**	**0.7405**
6	0.7324	0.5573	0.7558	0.8510	**0.9441**	**0.8336**	**0.8938**	**0.9430**
7	0.3731	0.1666	0.3175	0.4567	**0.4802**	**0.2557**	**0.3631**	**0.5227**
8	0.8720	0.8327	0.7712	0.8635	**0.9126**	**0.8971**	**0.8503**	**0.9161**
9	0.6725	0.6481	0.5596	0.6890	**0.6837**	**0.6573**	**0.5809**	**0.6961**
10	0.6210	0.6002	0.4560	0.6120	**0.6901**	**0.6740**	**0.5359**	**0.6873**
11	0.7977	0.7954	0.7303	0.8309	**0.8405**	**0.8236**	**0.7638**	**0.8557**
12	0.9579	0.9546	0.9221	0.9589	**0.9830**	**0.9821**	**0.9687**	**0.9839**
13	0.9123	0.9087	0.8525	0.9165	**0.9429**	**0.9412**	**0.9023**	**0.9468**
14	0.8851	0.8731	0.8097	0.8855	**0.9261**	**0.9175**	**0.8686**	**0.9251**
15	0.7821	0.7029	0.5857	0.6402	**0.9949**	**0.9935**	**0.9885**	**0.9936**

David L. Donoho and Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100 (10):5591–5596, 2003.

Danilo Samuel Jodas, Leandro Aparecido Passos, Ahsan Adeel, and João Paulo Papa. PL-*k* NN: A Parameterless Nearest Neighbors Classifier. In *2022 29th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 1–4, 2022.

Alexandre Luís Magalhães Levada, Frank Nielsen, and Michel Ferreira Cardia Haddad. Adaptive *k*-nearest neighbor classifier based on the local estimation of the shape operator. *arXiv preprint arXiv:2409.05084*, 2024.

Frank Nielsen. Supervised learning: practice and theory of classification with the *k*-NN rule. In *Introduction to HPC with MPI for Data Science*, pages 213–229. Springer, Switzerland, 2016.

Shahadat Uddin, Ibtisham Haque, Haohui Lu, Mohammad Ali Moni, and Ergun Gide. Comparative performance analysis of *k*-nearest neighbour (KNN) algorithm and its different variants for disease prediction. *Scientific Reports*, 12(1):6256, 2022.