Evaluation of KNN and SVM Algorithms

Vivek Raj

Department of Computer Science, Binghamton University

CS301: Ethical, Social, and Global Issues in Computing

Dr. George Weinschenk

November 5, 2021

Abstract

As data mining becomes an integral part of how humans use technology, understanding the different algorithms that power software applications becomes increasingly important. This paper compares two common data mining algorithms, KNN and SVM, in terms of their search time, model build time, and ability to accurately analyze data sets of different sizes. After examining how each algorithm functions under different conditions, the paper concludes that SVM is, overall, superior to KNN. While KNN serves as a strong algorithm for small data sets, an increase in the volume of unclassified data leads to an increase in KNN's search time and model build as well as a decrease in its accuracy. SVM, in contrast, performs well when analyzing large data sets and maintains a constant search time, low model build time, and high accuracy. This means that SVM is a better choice for complex data mining tasks including mining medical data sets, accurately predicting faces in facial recognition software, and completing bioinformatics tasks such as protein remote homology detection. SVM's superior data mining characteristics mean that this algorithm can contribute to social advancement in important ways by helping computer professionals analyze large amounts of information to identify patterns.

Evaluation of KNN and SVM Algorithms

Data mining, a relatively new technology in the field of computer science, is the process of separating and finding patterns in a comprehensive data set, involving methods such as machine learning, statistics, and database systems. Although data mining on medical data sets is particularly challenging, data mining in the medical field has increased significantly in the past few years due to mining's accuracy, ability to extract information from data, and ability to transform information into an understandable format for further use. For example, the ability to read complex medical data and classify it into groups helps doctors diagnose heart, cancer, and bladder-related problems.

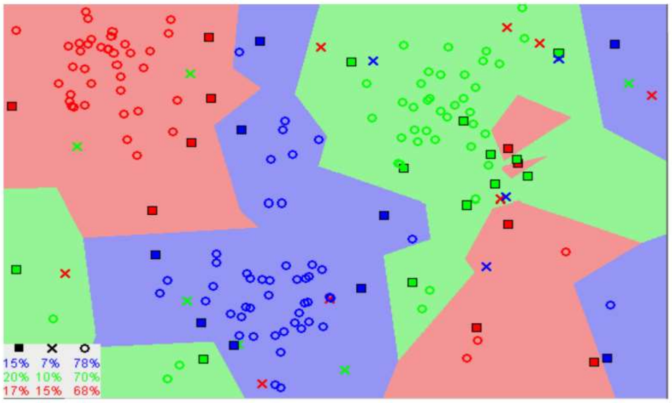
There are different classification algorithms that can be used when mining medical data sets, including K-nearest neighbors (KNN), decision tree, and support vector machine (SVM). When comparing two of these algorithms (SVM and KNN), it becomes clear that the SVM algorithm for machine learning is a better choice than the KNN classification algorithm because it performs better, has lower computational requirements, and provides more accurate and easier to interpret medical data sets, such as heart data sets, cancer datasets, and bladder volume monitoring. Some of these methods are soft in terms of rules and are suitable for weighted values. The following sections of this paper will discuss these methods in detail.

Alternative Technology: KNN

Supervised learning, also called supervised machine learning algorithms, uses different techniques to separate and inspect unlabeled data sets in a database system. These algorithms help identify hidden patterns or data sets without the need for human involvement or intervention. One of the supervised learning algorithms used to recognize patterns is the K-nearest neighbors (KNN). The KNN algorithm is a technique for classifying similar patterns in their proximity and does not depend on underlying data. In other words, similar things are close to each other in the problem space (Fig 1). KNN is a lazy learning algorithm that learns by storing the data it gets from the training set. When KNN gets new data, it classifies it into a similar category as the previous data sets. When KNN is in the training phase, it just stores the data; then, when it receives new data, it puts that data into a homogeneous category with the data it already has (Bhutani, 2021). In the KNN algorithm, an object is categorized by a majority vote of its neighbors. The object is allocated to the most common class among its closest K neighbors (k is a positive integer, generally small). If k = 1, then the object is allocated to its closest neighbor's class. Therefore, the algorithm hinges on this assumption being accurate enough to classify new cases based on a similarity measure (distance function/formula) (Bhutani, 2021).

Figure 1

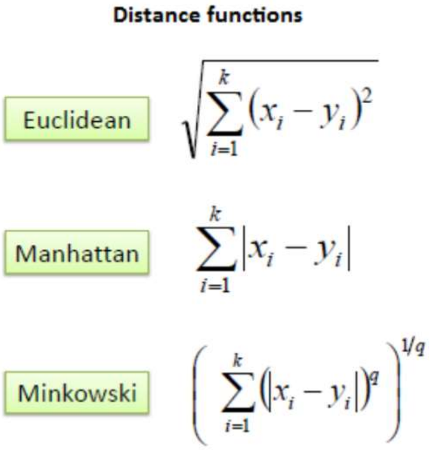
Data Points



Note: Images showing how similar data points typically exist close to each other (Harrison, 2018)

The distance function is used in the KNN algorithm to find the proximity of unclassified data to the classified data that is already present in the database. There are several ways to calculate the distance between classified and unclassified data sets, including Euclidean, Manhattan, or Murkowski (Fig 2).

Figure 2
Distance Functions


Distance functions
Euclidean
$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$
Manhattan
$\sum_{i=1}^{k}|x_i - y_i|$
Minkowski
$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

(Source: Sayad, n.d.)

Computationally demanding, the KNN algorithm slows as the volume of unclassified data added to the database increases (Bhutani, 2021). As complexity increases, interpreting data and using it in the medical field becomes more challenging. Hence, another useful supervised machine-learning algorithm for classification problems is Supported Vector Machine (SVM). Statistical-learning based SVM functions as one of the most robust methods for predicting data. Less complex and higher performing than KNN, SVM provides better data mining when used in medical data sets such as cancer datasets or heart datasets.

Support Vector Machine

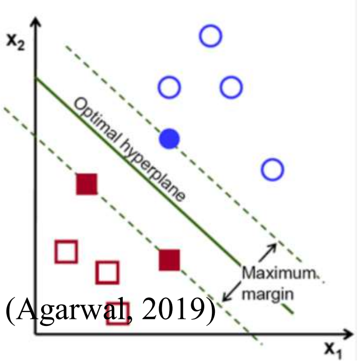
Vapnik proposed Support Vector Machine (SVM)—a supervised machine-learning algorithm used to classify data in the medical field—to solve two-class problems. SVM functions well in this field because it provides a user-friendly framework for data classification and understanding. SVM can classify data either linearly or nonlinearly. With a mathematical function capable of distinguishing two types of objects, SVM should not be confused with an implementation (a training SVM with the given data sets) (Raikwal & Saxena, 2012). In a

Support Vector Machine, one plots data values into n-dimensional space, where "n" stands for different types of characteristics, with the value of each character being the value of a coordinate. After this, one performs the classification by finding the hyper-plane that differentiates the two classes. The ultimate goal is to find a plane that can create the largest margin between the data points of the two classes (Agarwal, 2019).

To use SVM for solving a two-class problem, one places the sets in such a way in space that they are linearly or nonlinearly separable. Dealing with two classes, a drawing line method divides the data into two classes, shown in Fig. 1 and Fig. 2 (Kalcheva et al., 2020). The "hyperplane" (the separated data line) is chosen as far as possible from two data set classes, and (unlike with a K-NN algorithm), the SVM is trained using pre-classified documents.

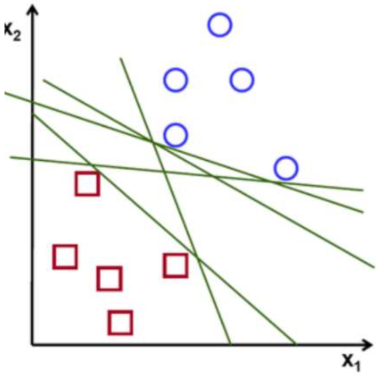
Figure 1

Possible Hyperplanes dividing two classes



(Agarwal, 2019)

Figure 2



The linear classification function f(x) is defined as:

F(x) =w^ (T) x+b -------------- (1)

In the above function, w^t is a weight vector and b is a deviation. To determine the classifier, one must find the w values. First, one finds the point that has the lowest deviation from the given data sets. For two linearly separable classes, the training data must satisfy the given condition (Nalepa & Kawulok, 2018).

$$\boldsymbol{w}^T\boldsymbol{x}_i + b \geq 1 \quad y_i = +1$$

$$\boldsymbol{w}^T\boldsymbol{x}_i + b \leq -1 \quad y_i = -1$$

The above two equations can also be written as:

$$Yi\ (w\hat{}T\ xi + b) - 1 \geq 0\ yi \in \{+1, -1\}\ \text{--------------}\ (4)$$

Sometimes, however, one encounters nonlinearly separable datasets; in such cases, one changes into linear separation by moving the data to another higher function space by transforming with a function of the input nonlinear data (Kalcheva et al., 2020). Therefore, one of the most famous formulas for classifying nonlinearly separable data is kernel function K, defined in the following equation (5):

k (xi,xj) = f(xi)f(xj) = K(L) = {V €V: L(v)=0}--------------(5)

In this equation, 0 denotes the null vector in W, and Kernel of L is a linear subspace of the domain V; the kernel of a linear operator's Rm – Rn is the same as the null space for the corresponding matrix (n x m). The kernel of a linear operator can sometimes be referred to by the operator's null space, while the kernel's dimension is often referred to as the operator’s nullity (Raikwal & Saxena, 2012). Other function formulas that one can use for nonlinear separable data sets are the radial basis function and the sigmoid function, as shown in Fig 3.

Figure 3

(Nalepa & Kawulok, 2018)

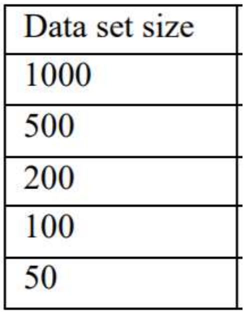
$$K(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

Implementation

Table 1 shows the implementation of the SVM and K-NN algorithms on the collected medical data sets of different (Raikwal & Saxena, 2012). One uses N cross-validation to evaluate the performance of both algorithms. According to the work of Raikwal (2012), performance analysis is done under accuracy, search time, and model build time. To obtain results, one will use five different datasets with different sizes as shown in Table 1 (Raikwal & Saxena, 2012).

Table 1

Showing Different Data Sets with Different Data Sizes



Data set size
1000
500
200
100
50

Results

The experiment results for these datasets are categorized into different parameters, which conveys an idea of how SVM outperforms KNN when the size of data sets increases.

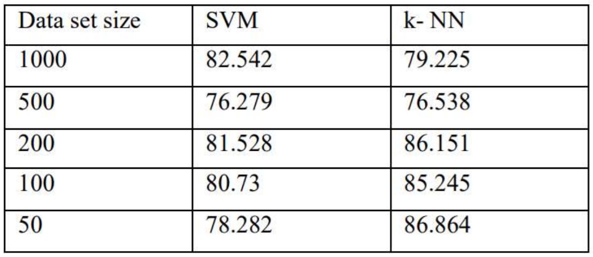
Accuracy

Accuracy is defined as the closeness of the prediction while using SVM and K-NN on data. For example, if the data contains 20 instances and 19 of them were found, one can say that the model is 95% accurate. The following formula calculates accuracy:

$$\text{Accuracy} = (\text{correct prediction} / \text{total supplied values})*100$$

Table 2

Showing the Accuracy of SVM and K-NN Systems



Data set size	SVM	k- NN
1000	82.542	79.225
500	76.279	76.538
200	81.528	86.151
100	80.73	85.245
50	78.282	86.864

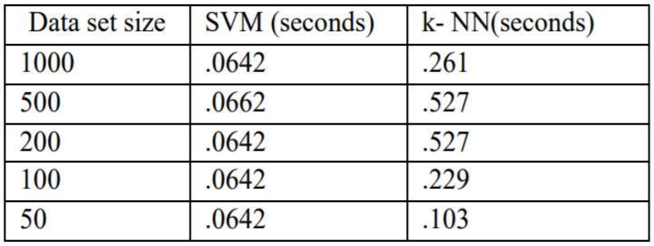
Table 2 demonstrates that K-NN accuracy surpasses SVM accuracy when the data set size is smaller; however, as the size of data increases, the accuracy for K-NN becomes much worse than the SVM algorithm (Raikwal & Saxena, 2012). Therefore, one can conclude that as the size of medical data sets (e.g. cancer data sets or bladder volume data sets) increases, the accuracy of the K-NN algorithm will decrease.

Memory

Search time, defined as the time required to find the predicting value from the given data sets, increases with K-NN (vs. SVM) when the size of the data set increases, as shown in Table 3 (Raikwal & Saxena, 2012). However, the search time for the SVM algorithm remains constant even if data size increases.

Table 3

Showing the Search Time Period for SVM and K-NN



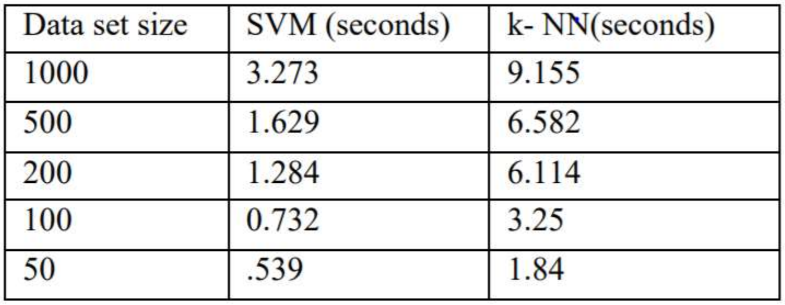
Data set size	SVM (seconds)	k- NN(seconds)
1000	.0642	.261
500	.0662	.527
200	.0642	.527
100	.0642	.229
50	.0642	.103

Model Build Time

Model build time is the time required for the algorithm to build and train itself from the given data set. Table 4 (Raikwal & Saxena, 2012) shows that K-NN always has a higher model build time than SVM, which means that K-NN will consume a lot of training time when the data set increases.

Table 4

Showing the Model Build Time for SVM and K-NN



Data set size	SVM (seconds)	k- NN(seconds)
1000	3.273	9.155
500	1.629	6.582
200	1.284	6.114
100	0.732	3.25
50	.539	1.84

The Social Impact of SVM: Real Life Applications

Computerized modeling and accurate prediction using data have become increasingly important in recent years, and machine learning languages SVM and K-NN have made this increasingly possible. However, SVM has proven more useful than K-NN because it makes fewer computational demands and is easier to interpret. Companies including Facebook, Google, and Uber as well as medical companies such as Pfizer, Sanofi, and Regeneron all use SVM and other machine learning as a central part of their operation for data prediction and classification (Team, 2021). More effective and user-friendly than other machine learning languages, SVM allows accurate classification of unseen information in fields such as facial recognition software and bioinformatics/medicine (West, 2019); in doing so, SVM has impacted society in important ways.

One socially significant application of SVM is in facial detection software. For example, face detection software uses SVM to classify images into “faces” and “non-faces.” It contains training data of pixels with two classes: face (+1) and non-face (-1). Then it extracts features from each pixel, creates a square boundary around faces based on pixel brightness, and classifies each image by using the same process (West, 2019). Kroeker (2019) notes that facial recognition software allows retailers to instantly identify when known shoplifters, organized retail criminals, or people with a history of retail fraud enter a shop. According to private university research, face recognition reduces external shrink by 34% and, more importantly, reduces violent incidents in retail stores that use face recognition software (Kroeker, 2009).

In addition to improving facial detection software, SVM has also impacted the fields of bioinformatics and medicine by helping address issues related to protein remote homology detection. By applying SVM calculations and algorithms to protein distant homology discovery, scientists can distinguish biological sequences (Team, 2021). For example, SVM can classify human genes and connect them to disease, thereby enabling scientists to more effectively find cures (Ray, 2021). This represents a tremendous social impact.

In both facial recognition and bioinformatics, SVM outperforms K-NN. When the K-NN algorithm deals with data, the accuracy of the algorithm's predictions depends on the quality of data that the system receives. In facial recognition software, this means that K-NN's performance will depend largely on the quality of the camera used during software development; if the pixel is not generated properly for the training dataset, then the prediction will differ from the actual result (Chatterjee, 2021). Similarly, accurate predictions of protein remote homology require enormous amounts of data, and K-NN's predictions slow when the size of the data set increases (Chatterjee, 2021). Moreover, K-NN requires a lot of memory to store training data, and such data storage can be expensive for a company or software developer (Chatterjee, 2021). SVM, on the other hand, has none of these problems which accompany K-NN.

When it comes to data accuracy, prediction time, and data storage, SVM consistently outperforms K-NN. Ethical concerns related to use of the SVM algorithm do exist—including privacy concerns and issues related to potential cloning and gene detection—but overall, the positive social impact of SVM far outweighs these risks.

Conclusion

Overall, both the KNN and SVM algorithms are useful for data mining, but these two algorithms have different strengths. KNN, a lazy learning algorithm, is more accurate for analyzing smaller data sets; however, as the volume of unclassified data increases, KNN's search time increases, its model build time increases, and its accuracy decreases. SVM, on the other hand, is an excellent choice for analyzing large data sets because its lower computational requirements allow it to maintain a constant search time even when data volume increases. Additionally, SVM has a lower model build time than KNN and maintains a high level of accuracy even when dealing with a large volume of unclassified data. This makes SVM superior to KNN when one needs to analyze high-volume data sets. SVM already demonstrates an ability to make a positive social impact: it is used for mining complicated data sets such as medical data sets, facial recognition software, and bioinformatics tasks such as remote protein homology detection. Software professionals should continue exploring new ways that the SVM machine learning algorithm can improve human lives.

References

Agarwal, A. (2019, October 3). *Support vector machine-formulation and derivation*. Medium. https://towardsdatascience.com/support-vector-machine-formulation-and-derivation-b146ce89f28

Bhutani, H. (2021 January 6). *K-Nearest neighbors (KNN) for machine learning*. Medium. https://medium.com/analytics-vidhya/k-nearest-neighbors-knn-for-machine-learning-d266d7c43830

Chatterjee, M. (2021, April 19). *A Quick Introduction to KNN Algorithm*. GreatLearning Blog: Free Resources What Matters to Shape Your Career! https://www.mygreatlearning.com/blog/knn-algorithm-introduction/

Gandhi, R. (2018, June 7). Towards data science. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

GeeksforGeeks. (2021, November 2). *K-Nearest Neighbours*. https://www.geeksforgeeks.org/k-nearest-neighbours/

Harrison, O. (2018, August 10). *Data points* [Image]. Towards Data Science. https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

Kalcheva, N., Karova, M., & Penev, I. (2020). Comparison of the accuracy of SVM kernel functions in text classification. *2020 International Conference on Biomedical Innovations and Applications (BIA)*, 141–145. https://doi.org/10.1109/BIA50171.2020.9244278

Kroeker, K. L. (2009, August 1). Face Recognition. Retrieved November 1, 2021, from https://dl.acm.org/doi/pdf/10.1145/1536616.1536623 .

Nalepa, J., & Kawulok, M. (2018). Selecting training sets for support vector machines: A review. *Artificial Intelligence Review*, *52*(2), 857–900. https://doi.org/10.1007/s10462-017-9611-1

Raikwal, J.S., & Saxena, K. (2012). Performance evaluation of SVM and K-nearest neighbor algorithm over medical data set. *International Journal of Computer Applications*, *50*(14), 35–39. https://doi.org/10.5120/7842-1055

Ray, S. (2021, August 26). *SVM | Support Vector Machine Algorithm in Machine Learning*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

Sayad, S. (n.d.) *Distance functions* [Image]. KNN Classification. https://www.saedsayad.com/k_nearest_neighbors.htm

Team, D. (2021, March 8). *Real-Life Applications of SVM (Support Vector Machines)*. DataFlair. https://data-flair.training/blogs/applications-of-svm

West, J. D. (2019, May 9). *21 Amazing Uses for Face Recognition – Facial Recognition Use Cases*. FaceFirst Face Recognition Software. https://www.facefirst.com/blog/amazing-uses-for-face-recognition-facial-recognition-use-cases/