COSC 528 Project 3: k-Nearest-Neighbors and Decision Trees

Elliot Greenlee

November 7, 2017

Abstract

This report covers the design and implementation of various nonparametric classification methods. k-Nearest-Neighbor and Decision Tree methods are used on medical data for breast cancer diagnosis. Additionally, dimensionality reduction by principal component analysis and hyperparameter optimization are applied for each method. From this analysis, it can be observed that kNN produces a best sensitivity of 96.1% at a k value of 8, and decision tree produces a best sensitivity of 97.6% with entropy as the impurity function and a maximum depth of 3. Using PCA to reduce the data to two components and then applying decision tree with a max depth of 2 and entropy as the impurity function produced the best results at 98.0% accuracy and 100% sensitivity, a tremendous result for a medical diagnostic tool.

1 Introduction

In Bayesian theory, one calculates the posterior probability of an input belonging to a particular class by taking the product of prior probability and likelihood of that event occurring and dividing it by the evidence. Methods based on this expression are a popular solution in pattern recognition.

$$P(w_j|x) = \frac{p(x|w_j)P(w_j)}{p(x)}$$

Classes are chosen based on the highest a-posteriori probability $P(w_j|x)$, calculated using the a-prior probability of a given class $P(w_j)$, the evidence $p(x)$, and the probability distribution function for the feature with respect to the given class $p(x|w_j)$.

There are two options for estimating the probability distribution of features: parametric methods and nonparametric methods. In parametric methods, the distribution is assumed to be known; for example, we assume a normal Gaussian distribution. In nonparametric methods, there is no assumption of the form of the distribution. In this project, k-nearest-neighbor and decision tree are used to estimate the distribution.

The objective of this project is achieve the highest possible accuracy on breast cancer diagnosis using two nonparametric classification methods: k-nearest-neighbor and decision tree. Dimensionality reduction by principal component analysis and hyperparameter optimization are also used to boost performance. Sections 2, 3, and 4 explain the methods used. Details of the experiments performed can be found in section 5. The work is summarize in section 6.

2 Dimensionality Reduction

Although in theory a greater number of statistically independent features should reduce further and further the error, in practice additional features often lead to worse performance. This occurs either when the wrong model is used, or because the training sample is not infinite, so the distributions are not accurately estimated. This leads to problems with overfitting, where the model works well on the training data but poorly on the testing data, and complexity, where the model takes longer to run at $O(d^2n)$ where d is dimension and n is number of samples. The solution to these problems is dimensionality reduction. Linear methods are used because of their simplicity, projecting high dimensional data onto a lower dimensional space.

One common method is principal component analysis, with the objective of minimizing information loss in a least-squares sense. Principal component analysis attempts to reduce the data while minimizing information loss. One might imagine the major and minor axis of a set of two dimensional elliptical data as in figure 1. Principal component analysis would choose the major axis as the projection direction to preserve information. Because there is no consideration of class information, it is an unsupervised method [2].

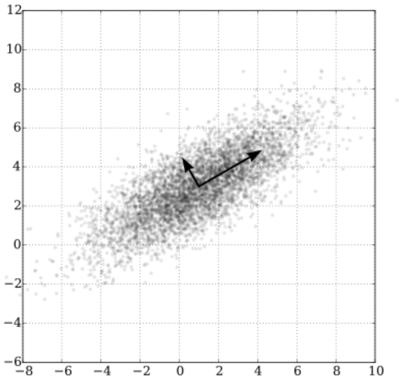

Figure 1: An example of the first two principal components on a graph.

One option for principal component analysis is singular value decomposition, which can decompose any Nxd matrix X into components VAW^T where V is the NxN matrix of the eigenvectors of XX^T in its columns, W is the dxd matrix of the eigenvectors of X^TX in its columns, and A is the Nxd matrix of the q singular values on its diagonal. For A, $q = \min(N, d)$, where the values are square roots of the nonzero eigenvalues of both XX^T and X^TX. The principal component P matrix of shape Nxd can be constructed as $P = (W^TX^T)^T$, where each row of P corresponds to one of the N instances. To reduce, take the first k columns of P [1].

3 k-Nearest-Neighbor

kNN is a nonparametric method that clusters the data. Clustering methods find groups of similar samples in the data as in figure 2 from researcher Kadir Peker.

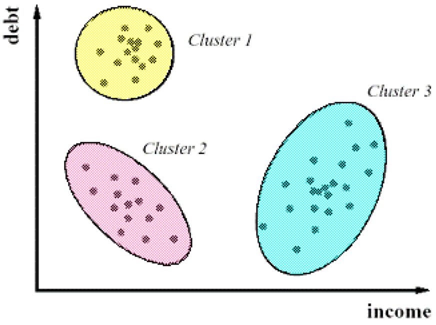

Figure 2: Income vs. debt cluster example

In this case, the discrimination is easy. With more data, more dimensions, and closer clusters, the problem becomes more challenging, and more appropriate for a computer to solve. Between the features of samples is a calculated measure of similarity, such as Euclidean distance. This metric can be used to compare samples, and to find clusters of similar samples [2].

In k-Nearest-Neighbor, for a specific point x, center a cell at the point and grow a hypersphere until it nearest encompasses k points. For this project euclidean distance is used. Bayes formula for the posterior probability of a sample being in a certain class is calculated as follows. We find the hypersphere of volume V which encloses k points from the set. if k_i of the points belong to class i, the density of class i can be estimated by:

$$P(C_i|x) = \frac{p(x|C_i)P(C_i)}{p(x)} = \frac{k_i}{k}$$

where $p(x|C_i) = k_i/(n_i V)$, $P(C_i) = n_i/n$, and $p(x) = k/(nV)$ If the probability of class i is more than j, choose class i, other wise class j. Ties are broken arbitrarily, and k is taken to be an odd number to minimize ties.

4 Decision Trees

In decision tree methods, a hierarchical model is used to identify recursive splits in attributes of the data. Each decision node of a decision tree functions as a split point on a threshold value. A given input is applied to the tree root node, and at each node is tested by a thresholding function, following a branch based on the result. Leaf nodes indicated regions in space where all instances fall into the same class. An example tree is shown in figure 3.

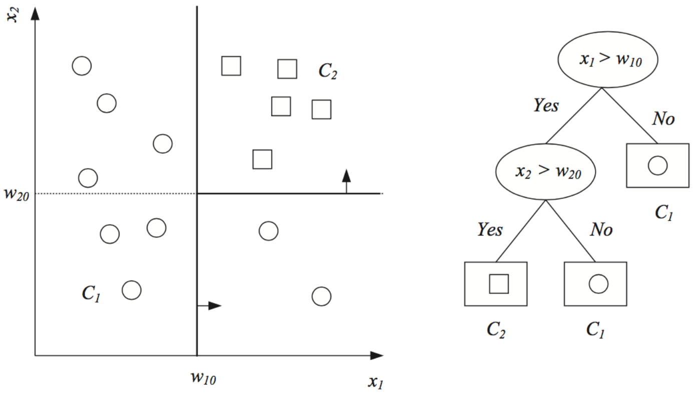

Figure 3: Example of a simple decision tree.

In the case of the Classification and Regression Tree (CART) implemented in this project, the tree is binary and used for classification. In this tree, each thresholding function works on a single attribute, sending values less than the threshold to the left node, and sending values greater than or equal to the threshold to the right. For a classification tree, training splits are decided based on an impurity measure. Splits are considered for all attributes and possible thresholds, and impurity is calculated for each, with the lowest impurity being selected.

For this project, three impurity measures are considered in the two class case, and are calculated as follows. For all methods the probability of one of the classes is

$$P(C_i|x,m) = p_m^i = \frac{N_m^i}{N_m}$$

Entropy is calculated as $\phi = -p\log_2 p - (1-p)\log_2(1-p)$. Gini Index is calculated as $\phi = 2p(1-p)$. Misclassification error is calculated as $\phi = 1 - \max(p, 1-p)$. If a node is not pure, it should be split to decrease impurity. In this project, splitting stops when a maximum tree depth is reached, a minimum impurity is reached, or when an impurity of zero is reached for a node.

5 Experiments and Results

These experiments are implemented in Python using the pandas, scipy, and scikit-learn libraries. For each accuracy measure, ten-fold cross validation is used to test the hyperparameters, and then the best methods are applied to the testing data.

5.1 Data

The data for this report is medical tumor features measured on a scale from 1 to 10; a class 2 for benign and 4 for malignant, is reported as the ground truth. The original data came as a csv file without headers, and missing 16 values in the Bare Nuclei column. The headers were added, and the rows containing missing data were removed. Additionally an index feature, Sample Code Number, was removed. This left 683 samples with 9 features each. The data is shown in figure 4 represented by two principal components, where the benign examples are colored green and the malignant examples are colored red. The minimum, maximum, mean, and standard deviation for each feature is reported in table 1 This information was provided as a csv file without source as part of COSC 528, Fall 2017 at the University of Tennessee, Knoxville [3].

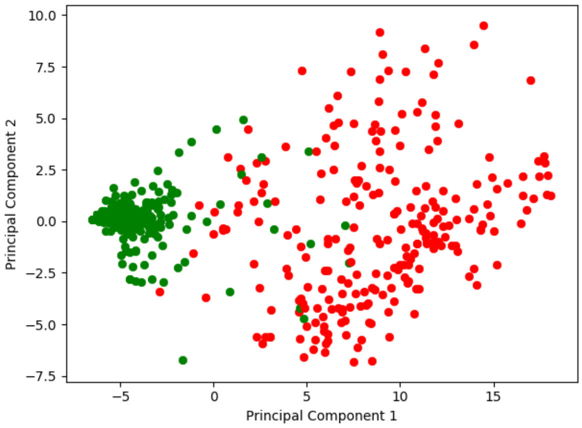

Figure 4: Dataset reduced to two principal components. Benign examples are shown in green and malignant examples are shown in red.

Feature	Minimum	Maximum	Mean	Standard Deviation
Clump Thickness	1	10	4.44	2.82
Uniformity of Cell Size	1	10	3.15	3.07
Uniformity of Cell Shape	1	10	3.22	2.99
Marginal Adhesion	1	10	2.83	2.86
Single Epithelial Cell Size	1	10	3.23	2.22
Bare Nuclei	1	10	3.54	3.64
Bland Chromatin	1	10	3.45	2.45
Normal Nucleoli	1	10	2.87	3.05
Mitoses	1	10	1.60	1.73
Class	2	4	2.70	0.95

Table 1: Minimum, maximum, mean, and standard deviation for each feature in the dataset.

5.2 Performance Metrics

For this project, multiple performance metrics are reported, including the confusion matrix, accuracy, sensitivity, precision, specificity, and f score of the predictions. Because this is a medical diagnosis, special consideration is given to the sensitivity of the method, which minimizes the number of false negatives. In a cancer diagnosis, a false negative can mean no treatment for someone with cancer, which is a more destructive outcome than treating someone without cancer. Overall, the methods with the maximum sensitivity are applied to the testing data, and each of the results below is reported for the best method with optimal hyperparameters. Note that the methods are trained to be accurate, not sensitive, so there is no issues with reporting all of the same class.

The confusion matrix reports true negatives tn, false negatives fn, false positives fp, and true positives tp. Accuracy is calculated as $(tn + tp)/(tn + tp + fn + fp)$. Sensitivity is calculated as $tp/(tp + fn)$. Precision is calculated as $tp/(tp + fp)$. Specificity is calculated as $tn/(tn + fp)$. The f score is calculated as $(precision * sensitivity)/(precision + sensitivity)$

5.3 k-Nearest-Neigbor

For these experiments, 80% of the data was used for cross validation, and 20% was used for testing. Metrics at various k values were calculated on the cross validation data; these are reported in table 2.

Metric	2	3	4	5	6	7	8	16	32
Accuracy	96.7	96.9	96.9	96.7	97.1	96.2	97.3	97.1	96.0
Sensitivity	96.8	94.4	95.7	93.8	96.3	93.8	96.9	95.0	91.9
False Positives	5	9	7	10	6	10	5	8	13

Table 2: Accuracy, sensitivity, and false negatives for various k values in kNN on the cross validation data.

The best results in sensitivity and accuracy came from a k value of 8. This implementation was applied to the testing data and returned an accuracy of 97.4%, a sensitivity of 96.1%, a precision of 96.8%, a specificity of 98.1%, an f score of 0.482, and the confusion matrix in 3.

	Predicted Class	
True Class	Benign	Malignant
Benign	211	4
Malignant	5	122

Table 3: Confusion matrix for kNN at a k value of 8.

5.4 Decision Tree

For these experiments, 50% of the data was used for cross validation, and 50% was used for testing. Metrics at various tree depths and impurity threshold limitations were calculated on the cross validation data; these are reported in tables 4 and 5. It is possible to see when the decision tree maxes out on accuracy; this occurs when the decision regions can no longer be split by axis aligned thresholds at the given depths.

Impurity Function	2	3	4	5	6	7	8	16	32
Entropy Accuracy	92.9	94.4	93.5	93.3	94.1	93.8	93.8	93.8	93.8
Entropy Sensitivity	94.6	94.6	92.9	90.1	91.1	90.2	90.2	90.2	90.2
Gini Index Accuracy	92.1	93.5	93.0	93.0	93.5	93.3	93.3	93.3	93.3
Gini Index Sensitivity	90.2	92.0	87.5	87.5	88.4	88.4	88.4	88.4	88.4
Misclassification Error Accuracy	93.0	93.0	93.8	93.3	92.7	92.7	92.1	91.8	91.8
Misclassification Error Sensitivity	90.2	91.1	91.1	88.4	85.7	85.7	84.8	83.9	83.9

Table 4: Accuracy and sensitivity data for various maximum depths of decision trees on the cross validation data, using three different impurity functions.

Impurity Function	0.3	0.1	0.03	0.01	0.003
Entropy	96.8/93.2	95.6/93.8	93.8/90.2	93.8/90.2	93.8/90.2
Gini Index	89.1/84.8	94.1/90.2	93.8/89.3	93.3/88.4	93.3/88.4
Misclassification Error	87.7/79.5	87.7/79.5	92.1/83.9	92.1/83.9	91.8/83.9

Table 5: Accuracy/sensitivity data for various impurity thresholds for decision trees on the cross validation data, using three different impurity functions.

The best results in sensitivity and accuracy came from a depth of three and an impurity of 0.1, both using entropy. The depth three implementation was applied to the testing data and returned an accuracy of 96.2%, a sensitivity of 97.6%, a precision of 92.5%, a specificity of 95.3%, an f score of 0.48, and the confusion matrix in 6.

The impurity 0.1 implementation was applied to the testing data and returned an accuracy of 94.7%, a sensitivity of 93.7%, a precision of 92.2%, a specificity of 95.3%, an f score of 0.46, and the confusion matrix in 7.

	Predicted Class	
True Class	Benign	Malignant
Benign	205	10
Malignant	3	124

Table 6: Confusion matrix for decision tree at a depth of 3 using entropy.

	Predicted Class	
True Class	Benign	Malignant
Benign	205	10
Malignant	8	119

Table 7: Confusion matrix for decision tree at an impurity threshold of 0.1 using entropy.

5.5 Dimensionality Reduction

Across all methods, entropy has the highest accuracies and sensitivities of the impurity functions even when controlling for other variables. For these experiments, 50% of the data was used for cross validation, and 50% was used for testing. Metrics at various tree depths were calculated on the cross validation data; these are reported in table 8.

PCs	2	3	4	5	6	7	8
1	96.5/97.3	95.6/96.4	94.7/93.8	94.4/92.0	94.4/91.2	94.7/92.9	94.1/91.8
2	96.5/98.2	96.2/97.3	94.7/92.9	95.0/92.9	94.7/92.9	94.7/92.9	95.0/93.8
3	96.5/98.1	95.9/93.8	95.6/93.8				
9	96.5/98.2	95.9/96.4	95.3/93.8	93.8/88.4	93.3/87.5	93.5/87.5	93.5/87.5

Table 8: Accuracy/sensitivity data for various maximum depths for decision trees on the PCA reduced cross validation data, with entropy as impurity. PC rows are the number of principal components.

The best results in sensitivity and accuracy came from reducing the data to the first two principal components and limiting the decision tree to a maximum depth of 2. This implementation was applied to the testing data and returned an accuracy of 98.0%, a sensitivity of 100%, a precision of 94.8%, a specificity of 96.7%, an f score of 0.49, and the confusion matrix in 9.

	Predicted Class	
True Class	Benign	Malignant
Benign	208	007
Malignant	000	127

Table 9: Confusion matrix for decision tree at a depth of 2 using entropy and 2 principal components.

6 Summary

In this project, k-Nearest-Neighbor and decision tree methods were applied to medical tumor data and PCA reduced data for breast cancer diagnosis. One of the more interesting findings in this project was that using entropy as the measure of impurity brought the best results across all methods, optimizations, and reductions. The best result for kNN was a sensitivity of 96.1% using a k value of 8. The best result for basic decision tree was a sensitivity of 97.6% using entropy and a max depth of 3. Using PCA to reduce the data to two components and then applying decision tree with a max depth of 2 and entropy as the impurity function produced the best results at 98.0% accuracy and 100% sensitivity, a tremendous result for a medical diagnostic tool.

The objectives of this project were to implement two nonparametric classification methods, and optimize them for a breast cancer diagnosis. From this, I learned the theory behind and implementation of decision tree models, something I have only ever used before now. I also became much more familiar with pandas dataframes and the sklearn library, tools I feel will be useful in my research. In future work, more visualization could be done, and other metrics could be considered instead of sensitivity.

References

[1] Ethem Alpaydin. *Introduction to Machine Learning Third Edition*. The MIT Press, 2014.

[2] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification Second Edition*. Wiley, 2001.

[3] Bruce MacLennan. Under 5 mortality rates 1800 to 2015, 2015. http://web.eecs.utk.edu/~mclennan/Classes/425-528/.

7 Appendix

The code for this project can be found on GitHub at https://github.com/LambentLight/knn-decisiontree. Please contact me with any questions or considerations.