Assignment 03
PHYS 555

Mukesh Aryal

02/25/2022

1 What is kernel trick in SVM and why do we use it? Explain few popular kernels in SVM!

Support vector machine (SVM) model is a supervised machine learning model which is mostly used for classification purposes; however, it can also be used for regression and outliers' detection. SVM uses a decision boundary to separate different classes of a dataset by observing the side of the decision boundary where the data lies on. The model creates a line, plane or a hyperplane in higher dimension to separate the data into different classes. When the dataset is complex and not linearly separable in n-dimension, then feature extraction can be used to add higher dimension to data where it may be possible to linearly separate the target classes. Adding many features to the data can increase the computational cost and can be very inefficient. To circumvent this problem, kernel trick is used.

The kernel trick represents the data only through a set of pairwise similarity comparisons between the original data in lower dimensional space instead of transforming and representing the data in higher dimensional feature space. In Kernel methods, the data is represented by an $n \times n$ kernel matrix of pairwise similarity comparisons where the entries (i, j) are defined by the kernel function: $k(x_i, x_j)$. The kernel function accepts inputs in the original lower dimensional space and returns the dot product of the transformed vectors in the higher dimensional space.

Some popular kernels in SVM are given below.

1. *Linear kernel*: It is the most basic type of kernel, usually one dimensional in nature, which works best when there are lots of features. The linear kernel is mostly preferred for text-classification problems.
   Kernel function : $F(x, x_i) = \sum(x.x_i)$

2. *Gaussian Radial Basis function (RBF)*: It is one of the most preferred and used kernel functions in svm suitable for non-linear data. It helps to make proper separation when there is no prior knowledge of data. Kernel function: $F(x, x_i) = \exp(-\gamma||x - x_i||^2)$

3. *Sigmoid kernel*: This is mostly preferred for neural networks. This kernel function is similar to a two-layer perceptron model of the neural network, which works as an activation function for neurons. Kernel function: $F(x, x_i) = tanh(\gamma.x^T x_i + r)$

4. *Gaussian kernel*: It is a commonly used kernel. It is used when there is no prior knowledge of a given dataset. Kernel function: $k(x, x_i) = exp(\frac{||x-x_i||^2}{2\sigma^2})$

2 How can SVM be used for regression? Explain in detail!

The same principle of SVM used for classification can be used for regression. When doing classification, SVM model uses the hard boundary and soft margins to divide different classes of dataset minimizing the number of data points to lie between the soft and hard margins. For regression, SVM uses the soft margins to confine the data points in between the soft margins or the tube. By doing so, SVM approximates the equation of the hyperplane as the best fit for the dataset. An example implementation of support vector regression using linear SVR is illustrated in figure 01.

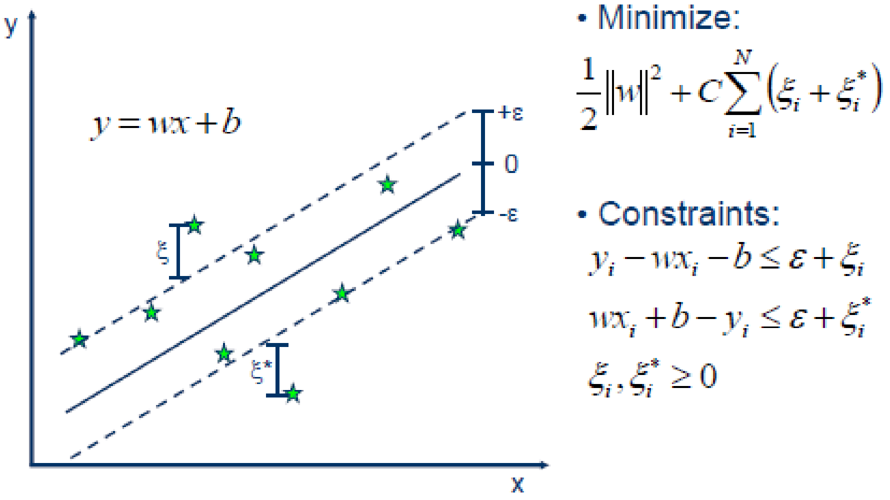
y
$y = wx + b$
+ε
0
-ε
ξ
ξ*
x
• Minimize:
$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$
• Constraints:
$y_i - wx_i - b \leq \varepsilon + \xi_i$
$wx_i + b - y_i \leq \varepsilon + \xi_i^*$
$\xi_i, \xi_i^* \geq 0$

Figure 01: Structure of SVM

The distance between the hyperplane and soft margin is ϵ, a margin of tolerance. SVR tries to determine the hyperplane by minimizing the range between the predicted and observed values. Minimizing the value of w in the equation shown in the figure is similar to the value defined to maximize the margin. The summation part in the equation used by SVR to obtain best fit represents an empirical error. To minimize the error, equation (1) with Kernel function is used.

$$f(x) = \sum_{i}^{n}(\alpha_i^* + \alpha_i)K(x, x_i) + B \tag{1}$$

Here the alpha term represents the Lagrange multiplier, and its value is greater than or equal to 1. K represents the kernel function and B represents the bias term.

In this fashion, SVM can be used for regression purposes. It predicts and performs better than other models like KNN and linear regression because it includes improved optimization strategies for a broad set of variables.

3 What is kernel PCA, and why could it be effective? Explain in detail!

Kernel PCA is an extension of principal component analysis that utilizes kernel methods and enables nonlinear dimensionality reduction. Since PCA transformation are linear transformations, it works best for datasets which are linearly separable. If the dataset is non-linear then using PCA on the dataset does not yield optimal dimensionality reduction. Kernel PCA gets around this problem by using a kernel function to project dataset into a higher dimensional feature space where the dataset is linearly separable first then uses PCA on the dataset to obtain best dimensionality reduction. The effectiveness of kernel PCA over ordinary PCA is explained using the example below.

Consider a classification problem using PCA where the decision boundary in original lower dimension is non-linear. Suppose the decision boundary for the dataset in original dimension is a third order polynomial, $y = a + bx + cx^2 + dx^3$. The function when plotted in the usual x-y plane will produce a non-linear wavy line. We can do feature extraction and go to higher dimension adding square and cubic terms. By increasing the features and utilizing them, the decision boundary then becomes a hyperplane. By doing so, we can recover the linearity of the boundary and do the usual PCA decomposition on the dataset.

Adding extra features to an already large dataset results in large volume of variables and significantly increase the computational cost. PCA utilizing the kernel method circumvent this problem. If $\mathbf{x}$ is the set of n variables, and $\phi(\mathbf{x})$ represents the non-linear combination of these variables into a $m > n$ dataset. After computing the kernel function, $k(\mathbf{x}) = \phi(\mathbf{x})\phi^T(\mathbf{x})$, the kernel function can play the same role as the covariance matrix did for linear PCA. In this way, the eigenvalues and eigenvectors of the kernel matrix can then be used as the principal components of the m-dimensional space where original variables were mapped into. The kernel trick when added to PCA prevents the number of variables to blow up and successfully captures the non-linearity of the dataset. Therefore, kernel PCA can be very effective for dealing with non-linear dataset and their optimal dimensionality reduction.

4 Explain XGBoost method and compare it with Random Forest. List pros and cons of each approach!

XGBoost, Extreme Gradient Boosting, is a scalable and distributed gradient-boosted decision tree (GBDT) model that provides parallel tree boosting. A gradient boosting decision tree is a decision tree similar to random forest that can be used for classification and regression. Both random forest and GBDT build a model consisting of multiple decision trees; however, the trees are built and combined differently by both methods. Random forest uses a bagging technique to build decision trees in parallel from random bootstrap samples of the data set. It then averages the predictions of all decision trees and uses the average as the final prediction.

Boosting refers to the idea of improving a single weak model by combining it with a number of other weak models to generate a collectively strong model. Gradient boosting is an extension of boosting where the process of additively generating weak models is formalized as a gradient descent algorithm over an objective function. Gradient boosting sets targeted outcomes for the next model in an effort to minimize errors. Targeted outcomes for each case are based on the gradient of the error with respect to the prediction. GBDTs iteratively train an ensemble of shallow decision trees, with each iteration using the error residuals of the previous model to fit the next model. The final prediction is a weighted sum of all the tree predictions.

Random forest "bagging" minimizes the variance and overfitting, while GBDT "boosting" minimizes the bias and underfitting. XGBoost is a scalable and highly accurate implementation of gradient boosting where trees are built in parallel instead of sequentially making them like GBDT. Advantages and disadvantages of both methods are listed below.

Random Forest

Pros:

1. Robust to outliers.
2. Works well with non-linear data.
3. Lower risk of overfitting.
4. Runs efficiently on a large dataset.
5. Better accuracy than other classification algorithms.
6. Good performance on imbalanced datasets.

Cons:

1. It is biased while dealing with categorical variables.
2. Slow training.
3. Not suitable for linear methods with a lot of sparse features.

XGBoost

Pros:

1. Requires less feature engineering.

2. It outputs importance of each feature.
3. Fast to interpret.
4. Robust to outliers.
5. Handles large size datasets well.
6. Good execution speed.
7. Less prone to overfitting.

Cons:

1. Harder to interpret and visualize.
2. Prone to overfitting if parameters not tuned properly.
3. Harder to tune as there are too many hyperparameters.

5 What is a Gaussian Mixture method? How does it work? Describe some of its application in detail!

Gaussian mixture model (GMM) is an unsupervised machine learning model that is used to discover and cluster the underlying groups of data. GMM assumes the mixed data of different clusters are distributed normally and they have means and covariances. In higher feature space, GMM consists of two parameters: mean vectors (μ) and covariance matrices (Σ). The parameters and example distribution utilizing GMM is given in figure 02.

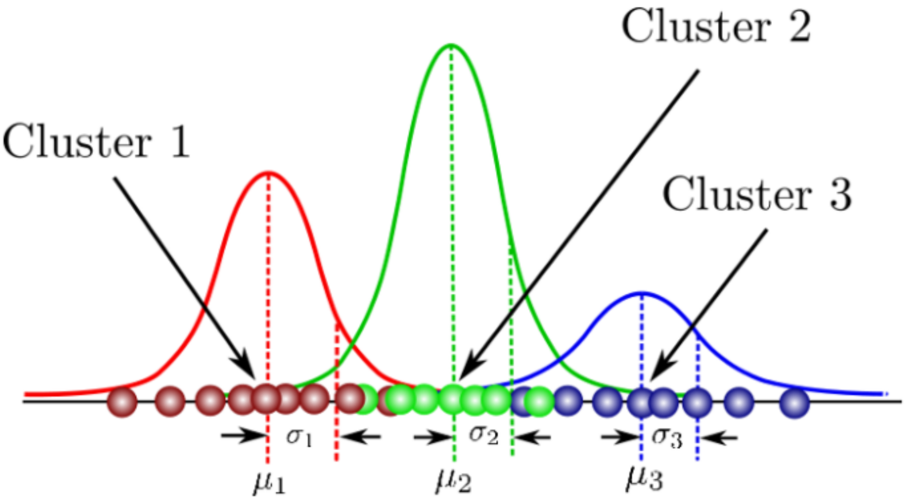

Figure 02: Parameters of GMM

In GMM, a so-called expectation-maximization (EM) method is used to find the gaussian mixture model parameters. Expectation step is used to find the gaussian parameters which are used to represent each component of gaussian mixture models. Maximization step is used to determine whether new data points can be added to the cluster. EM method is an iterative algorithm that alternates between expectation step and maximization step. The iterative process first computes expectations for each data point using current parameter estimates and then maximizes them to produce new gaussian. Later gaussian means are updated based on the maximum likelihood estimate and the two-step cycle is repeated until the gaussian parameters converge.

Some applications where GMM can be used are discussed below.

1. Medical dataset: GMMs can be used to find specific patters in medical dataset by segmenting images into multiple categories based on their content. Patters can be cancers or abnormality.

2. Modeling natural phenomena: Natural phenomena that includes noises that are distributed normally can be modeled using GMM.

3. Stock price prediction: GMM can be used in finance to detect change points in time series data that helps to find turning points of stock prices or other market movements which are difficult to spot due to noise and volatility.

4. Gene study: GMM can be used to detect different genes responsible for two different conditions and identify the types of genes contributing to a certain disease state.