Support Vector Machines

Ilmari Vahteristo (1107891), Yahya Makhlouf (1107658)

June 6, 2023

Contents

1 Introduction

The goal of this project was to get familiar with the **Support Vector Machine** (SVM) algorithm, a supervised learning method commonly used for regression and classification. We introduce the theory behind SVM (mainly for classification) and kernel methods and apply them to a news headline sentiment classification problem by creating word embeddings using a pre-trained Word-To-Vector model. We will only briefly discuss the algorithm behind word embeddings to keep this project shorter.

2 Kernel methods

Classic classification and regression model linear or not, consider a mapping from **y(x,w)** in which from an input **x** we output **y** using a vector of adaptive parameters **w**. The training data in these models is merely used from training and is disregarded when making the predictions.
However, with the model using kernel methods, often called *memory-based methods*, the training data is used for prediction whether it's a regression problem or classification problem. In these models, our goal is to make predictions based on similarities between the input in training and testing. The similarity is measured with a function called **kernel function**.

2.1 Kernel functions

Linear parametric models can be expressed as a dual representation where predictions are made by combining a kernel function evaluated at the training data points. This applies to models that rely on a fixed nonlinear feature space mapping $\phi(x)$, where the kernel function is determined by :

$$k(\boldsymbol{x}, \boldsymbol{x'}) = \phi(\boldsymbol{x})^{\boldsymbol{T}} \phi(\boldsymbol{x'})$$

As we can see kernel functions are symmetric and has a notion of "product". It can either be constructed from the kernel substitution function $\phi(x)$ or by combination and/or composition of other valid kernel functions.

2.2 Classification with a Gaussian Process

In a probabilistic approach to classification, our aim is to create a model that can estimate the posterior probabilities of the target variable for a new input vector, based on a given set of training data. Since Gaussian process output lie on the entire real-axis, we can use a nonlinear activation function to adapt it to (0,1).
Thus, let's consider a Gaussian process with the function f($\boldsymbol{x}$) adjusted with a sigmoid σ where the output y = σ(f)$\in (0, 1)$

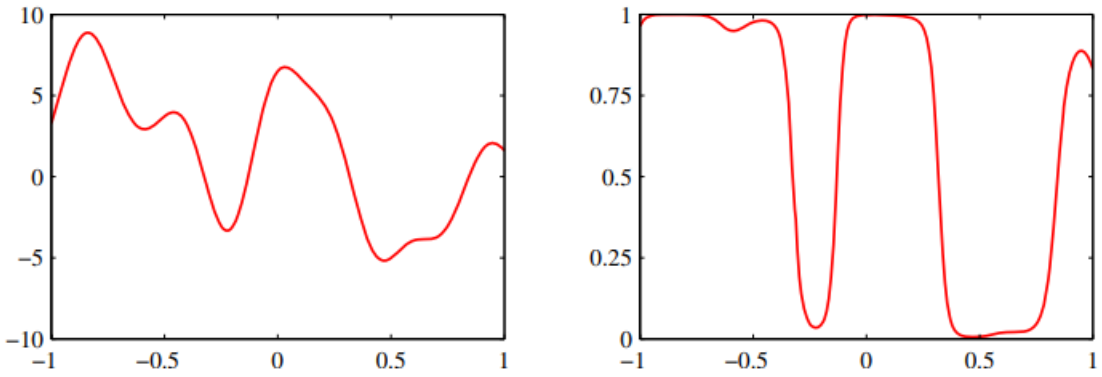

Figure 1: Transforming a sample from a Gaussian Process using a logistic sigmoid function.

Example : Let's consider a training set of of inputs $\mathbf{X} = (x_i)_{i\in[1,N]}$ with observed targets $\mathbf{Z} = (z_i)_{i\in[1,N]} \in (0, 1)$.

For a test point (x',z'), our goal will be predict p(z'—x',X,Z), since this example is a simple two-class problem (0,1), p(z' = 1 —x',X,Z) will suffy.
The value p(z' = 0 —x',X,Z) can be obtained with p(z' = 0 —x',X,Z) = 1- p(z' = 1 —x',X,Z).
The Gaussian process prior for $F = (f(x_i), f(x'))_i \in [1, N])$ can be expressed as

$$p(F) = \mathcal{N}(F|0, C), \ \ and \ \forall x_i, x_j \ C(x_i, x_j) = k(x_i, x_j) + \delta_{ij}\alpha$$

Where is the δ Kronecker symbol, α a regulating parameter fixed in advance and the covariance matrix C expressed using a positive semi definite kernel function k. Finally we obtain our predicted distribution :

$$p(z' = 1|x', X, Z) = \int p(z' = 1|F)p(F|x', X, Z)dF$$

3 Support Vector Machine

SVMs are a class of algorithms often used for classification, regression, and outlier detection. They are fairly memory efficient, robust, and effective in high-dimensional spaces even if the number of features is smaller than the number of samples.
Given training data pairs (X_i, y_i) where X_i is a d dimensional feature vector and $y_i \in -1, 1$ (binary classification), SVM finds two parallel hyperplanes which best separate the two classes. The distance between the hyperplanes is called the **margin**.

3.1 Hard margin

If the data can be separated by two parallel planes, such that no point lies in the margin, then we can use SVM with a hard margin. With a hard margin, we find the

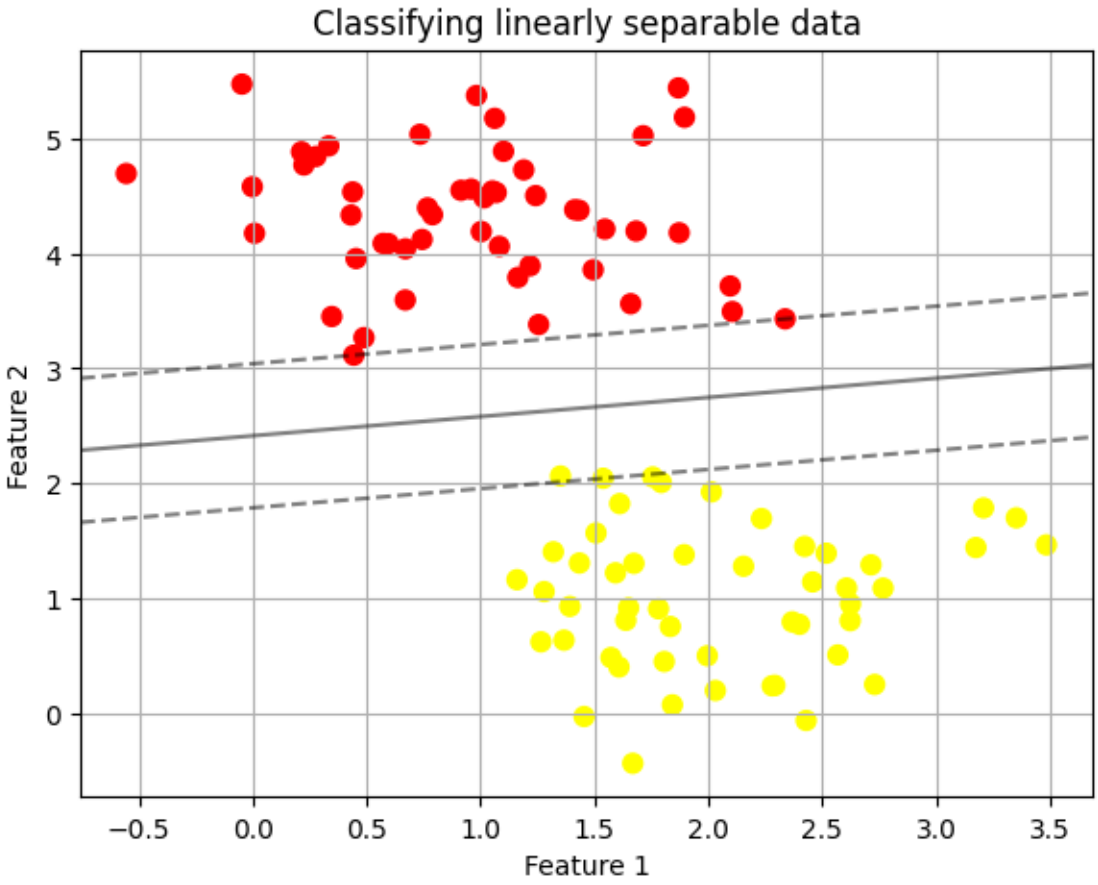

Figure 2: SVM finding a hard-margin between two classes in 2D, with artificial data.

two hyperplanes that have the largest distance between them.
The two hyperplanes have the form:

$$\boldsymbol{w^T * x - b = -1 \; or \; 1}$$

, where any x where the answer is above 1 is classified as a 1, and vice versa.

The distance between the planes can be calculated, for example, by using an orthogonal projection, which comes out to be $\frac{2}{||w||}$. To maximize the margin, we must minimize $||w||$. And since no point can be in the margin, we have two constraints from the above function.
The points closest to the margin are called the support vectors.

3.2 Soft margin

In most real-life cases, the classes are probably not clearly separable, so a more general method is needed, which allows points to be in the margin.

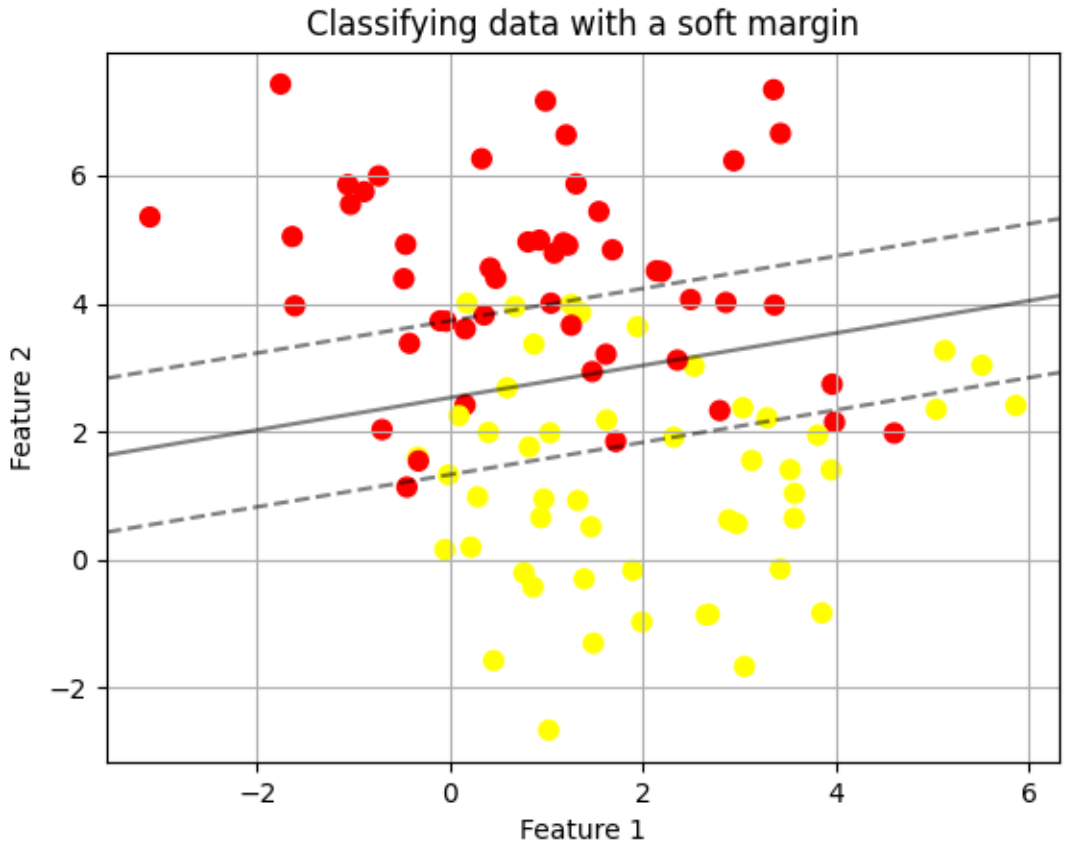

Figure 3: A linear classifier using a soft margin to separate the artificial data.

To allow misclassifications, a hinge loss function is typically used:

$$l(\hat{y}) = max(0, 1 - t * \hat{y})$$

where t is desired output $-1, 1$ and $\hat{y}$ is the classifiers prediction. So in the case of SVM $y = \boldsymbol{w}^{\boldsymbol{T}} * \boldsymbol{x} - b$.
Furthermore, a parameter λ which controls the effect of the size of the margin is added to the minimization problem, now described by:

$$\lambda * ||\boldsymbol{w}||^2 + \frac{1}{n} \sum_{i=1}^{n} l(\boldsymbol{w}^{\boldsymbol{T}} * \boldsymbol{x} - b)$$

where n is the sample size.

4 Application

To test the performance of SVMs and to create a useful application, we wanted to predict whether an investor considers a news headline good, neutral, or negative for the stock/market. For training the model, we obtained a classification dataset, containing 4800 financial news headlines along with their labels (https://www.kaggle.com/datasets/ankurzing/sent analysis-for-financial-news).
Since SVM doesn't understand strings (at least without a proper kernel function), we used a pre-trained word-to-vector model (https://huggingface.co/fse/word2vec-google-news-300) to turn the headlines of the dataset into 300-dimensional vectors.
To optimize parameters for the SVM classifier, we ran a grid search on the parameters and tested different kernels, label-specific λ regularization parameters, and different under-/oversampling techniques due to the imbalance in the data. We selected the model with the highest unbalanced (macro) F1 score.
Finally, we tested the model on randomly selected 20% of the data. The model achieves a 75.6% accuracy and an F1 score (macro) of 0.72. The accuracy results we obtain are similar to the results the original authors achieved.
Here are also predictions for some real or made-up news headlines:

- *US banks prepare for losses in rush for commercial property exit.*: negative
- *Oil prices pop after Saudi Arabia pledges more voluntary production cuts.*: negative
- *Nvidia short-sellers bleed $3.6bn in May as AI boom continues.*: negative
- *Silicon Valley's tech giants are in trouble. Here's why.*: negative
- *Now is a good time to buy stocks, history shows.*: neutral
- *Federal Reserve's Jerome Powell says economic recovery could stretch through end of 2023.*: neutral
- *Putin moves to extend his rule until 2036 after Russians vote to back constitutional changes.*: neutral
- *U.S. stock futures show SP 500 consolidating at 2023 highs after Friday's powerful session.*: positive
- *Amazon beats expectations with $88.9bn in sales, stock surges.*: positive
- *Google issues positive earnings surprise as ad revenue rebounds.*: positive

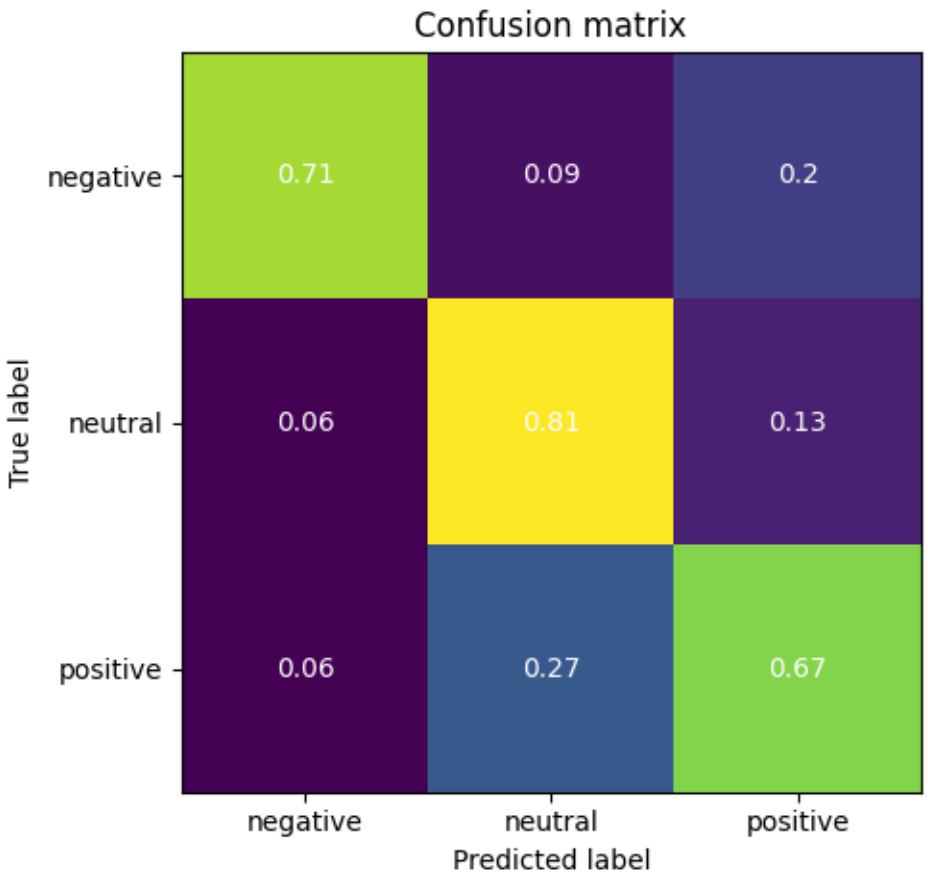

Figure 4: The confusion matrix of the SVM classifier.