Support Vector Machines

Tirthankar Mazumder Parth Laturia Pratip Dalal

Indian Institute of Technology, Bombay

Spring 2022

Outline

History of SVMs

- Developed at Bell Laboratories by Vladimir Vapnik with colleagues, in a series of papers (Boser et al., 1992, Guyon et al., 1993, Cortes and Vapnik, 1995, Vapnik et al., 1997)

History of SVMs

- Developed at Bell Laboratories by Vladimir Vapnik with colleagues, in a series of papers (Boser et al., 1992, Guyon et al., 1993, Cortes and Vapnik, 1995, Vapnik et al., 1997)
- At its core, support vector machines (aka support vector networks) is a classification algorithm.

History of SVMs

- Developed at Bell Laboratories by Vladimir Vapnik with colleagues, in a series of papers (Boser et al., 1992, Guyon et al., 1993, Cortes and Vapnik, 1995, Vapnik et al., 1997)
- At its core, support vector machines (aka support vector networks) is a classification algorithm.
  - Given a set of training examples, each marked as belonging to one of two categories, the SVM algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

History of SVMs

- Developed at Bell Laboratories by Vladimir Vapnik with colleagues, in a series of papers (Boser et al., 1992, Guyon et al., 1993, Cortes and Vapnik, 1995, Vapnik et al., 1997)
- At its core, support vector machines (aka support vector networks) is a classification algorithm.
  - Given a set of training examples, each marked as belonging to one of two categories, the SVM algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.
- In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is known as the kernel trick.

History of SVMs

- Developed at Bell Laboratories by Vladimir Vapnik with colleagues, in a series of papers (Boser et al., 1992, Guyon et al., 1993, Cortes and Vapnik, 1995, Vapnik et al., 1997)
- At its core, support vector machines (aka support vector networks) is a classification algorithm.
  - Given a set of training examples, each marked as belonging to one of two categories, the SVM algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.
- In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is known as the kernel trick.
  - The kernel trick involves implicitly a clever generalization of the Euclidean inner product $\langle \cdot, \cdot \rangle$: We use the new kernel $k(\mathbf{x}, \mathbf{y}) = \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}) \rangle$ where the new kernel is a proper inner product.

Motivation

- Classifying data is a common task in many problem domains; machine learning in particular.

Motivation

- Classifying data is a common task in many problem domains; machine learning in particular.
- Suppose that, given some data points which belong in one of two classes, the task is to determine which class a new data point will be in.

Motivation

- Classifying data is a common task in many problem domains; machine learning in particular.
- Suppose that, given some data points which belong in one of two classes, the task is to determine which class a new data point will be in.
- In the case of SVMs (in the simplest case), a data point is a vector in $\mathbb{R}^n$, and the task is to determine a $(n-1)$ dimensional hyperplane which separates the two classes of data points.

Motivation

- Classifying data is a common task in many problem domains; machine learning in particular.
- Suppose that, given some data points which belong in one of two classes, the task is to determine which class a new data point will be in.
- In the case of SVMs (in the simplest case), a data point is a vector in $\mathbb{R}^n$, and the task is to determine a $(n-1)$ dimensional hyperplane which separates the two classes of data points.
- However, there are many hyperplane which classify the given data, so it is natural to wish to find the "best" one for some suitable definition of "best".

Motivation

- Classifying data is a common task in many problem domains; machine learning in particular.
- Suppose that, given some data points which belong in one of two classes, the task is to determine which class a new data point will be in.
- In the case of SVMs (in the simplest case), a data point is a vector in $\mathbb{R}^n$, and the task is to determine a $(n-1)$ dimensional hyperplane which separates the two classes of data points.
- However, there are many hyperplane which classify the given data, so it is natural to wish to find the "best" one for some suitable definition of "best".
- One reasonable choice as the "best" hyperplane is the one that represents the largest separation between the two classes. That is,we choose the hyperplane with the distance from it to the nearest data point on each side.

Motivation contd.

- Whereas the original problem may be stated in a finite-dimensional space, it often happens that the sets to discriminate are not linearly separable in that space.

Motivation contd.

- Whereas the original problem may be stated in a finite-dimensional space, it often happens that the sets to discriminate are not linearly separable in that space.
- For this reason, we map the original finite-dimensional space into a much higher-dimensional space to make the separation easier.

Motivation contd.

- Whereas the original problem may be stated in a finite-dimensional space, it often happens that the sets to discriminate are not linearly separable in that space.
- For this reason, we map the original finite-dimensional space into a much higher-dimensional space to make the separation easier.
- To keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that dot products of input data vectors can be easily computed in terms of the variables in the original space, by defining them in terms of a kernel functions (using the aforementioned kernel trick) selected to suit the problem.

Applications

- Classification of satellite data

Applications

- Classification of satellite data
- Text and hypertext categorization

Applications

- Classification of satellite data
- Text and hypertext categorization
    - Text and hypertext categorization is the problem of assigning one or more categories to a body of text (for example, fiction vs non-fiction, biography, novel, etc.)

Applications

- Classification of satellite data
- Text and hypertext categorization
  - Text and hypertext categorization is the problem of assigning one or more categories to a body of text (for example, fiction vs non-fiction, biography, novel, etc.)
- Image Classification

Applications

- Classification of satellite data
- Text and hypertext categorization
  - Text and hypertext categorization is the problem of assigning one or more categories to a body of text (for example, fiction vs non-fiction, biography, novel, etc.)
- Image Classification
  - Experimental results show that SVMs achieve significantly higher search accuracy than traditional query refinement schemes after just three to four rounds of relevance feedback.

Applications

- Classification of satellite data
- Text and hypertext categorization
    - Text and hypertext categorization is the problem of assigning one or more categories to a body of text (for example, fiction vs non-fiction, biography, novel, etc.)
- Image Classification
    - Experimental results show that SVMs achieve significantly higher search accuracy than traditional query refinement schemes after just three to four rounds of relevance feedback.
- Handwriting Recognition

Applications

- Classification of satellite data
- Text and hypertext categorization
  - Text and hypertext categorization is the problem of assigning one or more categories to a body of text (for example, fiction vs non-fiction, biography, novel, etc.)
- Image Classification
  - Experimental results show that SVMs achieve significantly higher search accuracy than traditional query refinement schemes after just three to four rounds of relevance feedback.
- Handwriting Recognition
- Wind Speed Prediction

Hard Margin SVM

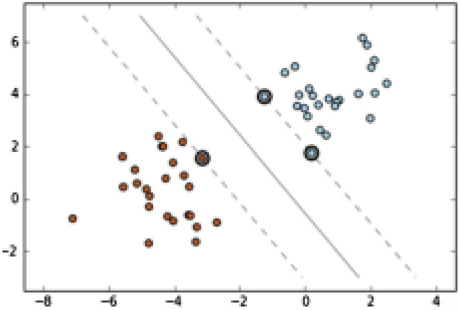


Figure: Caption

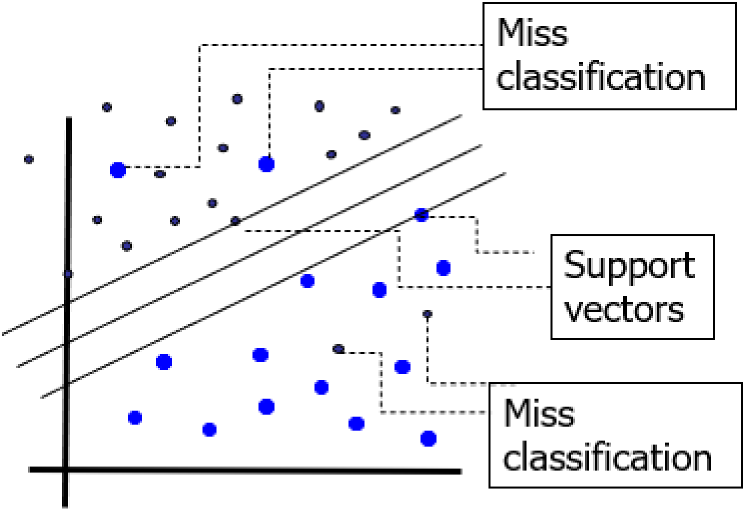
Hard SVM Algorithm

Suppose the training set of instance-label pairs is denoted by x_i, y_i, for $i \in \{1, 2, \ldots, \ell\}$ where $x_i \in \mathbb{R}^n$, $y_i \in \{+1, -1\}$, n is the number of features and ℓ is the number of training data points. Then hard margin linear SVM solves the following optimization problem:

$$\min_{w,b} \ \frac{1}{2}\|w\|^2$$
$$\text{s.t. } y_i(w^T x_i + b) \geq 1 \quad \forall i$$

where w, b are parameters.

Soft margin SVM



Soft SVM Algorithm

Soft margin SVM solves the optimization problem:

$$\min_{w,b} \ \frac{1}{2}(\|w\|^2) + C\sum_{i=1}^{n} x_i$$

$$s.t.\ y_i(w^T x_i + b) \geq 1 - t_i, t_i \geq 1\ \forall i$$

where t_i is a slack variable to allow mis-classification and C is a penalty parameter.

Other Proposed SVM

- **One-versus-rest (1998):** One of the mostly used multi-class SVM. In this technique, we solve *m* binary SVM problems where *m* is the number of classes.
- **Transductive SVM (1999):** The goal is to classify a given dataset with as few errors as possible without caring about the particular decision function.
- **Least squares SVM (1999):** Proposes a least squared version of SVM with equality constraints. Due to equality constraints, it solves a set of linear equations to find the solution, unlike standard SVM which solves quadratic programming.
- **Proximal SVM (2001):** Classifies data points on the basis of proximity to the two parallel planes.

Other SVM contd.

- **Fuzzy SVM (2002):** In this SVM the contributions of points are used to find the separating hyperplane. FSVM is useful in reducing the effect of outliers and noise. It is suitable for applications where data points have modeled characteristics.
- **Kernel SVM:** For non-linearly separable data, soft margin classifiers does not generalize well and produces a lot of mis-classification errors. In such cases, Kernels are used, which transform the data from the Input Space to the Feature Space where the data is linearly separable.

Optimal margin classifier

Formulation of the optimization problem:

$$\min_{w,b} \ \frac{1}{2}\|w\|^2 s.t. \ y_i(w^T x_i + b) \geq 1 \quad \forall i \tag{1}$$

The Lagrangian function of the optimization problem:

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(w^T x_i + b) - 1) \tag{2}$$

where α_i are the Lagrange multipliers.

Dual form of problem

$$\nabla_w L(w, b, \alpha) = w - \sum_{i=1}^{n} \alpha_i y_i x_i = \mathbf{0}$$

which implies that

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{3}$$

The derivative w.r.t b gives:

$$\nabla_b L(w, b, \alpha) = \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{4}$$

$$L(w, b, \alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j (x_i)^T x_j - b \sum_{i=1}^{n} \alpha_i y_i \tag{5}$$

Dual form of optimization problem

Using the equation (4) we have

$$L(w, b, \alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j (x_i)^T x_j \tag{6}$$

Thus the dual form of the problem becomes:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j (x_i)^T x_j$$

$$s.t. \ \alpha_i \geq 0 \ \forall i \in \{1, 2, \ldots, n\}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

Finding optimal w^* and b^* and model flexibility

- The dual formulation of the problem gives optimal α^*
- Corresponding to optimal α^* optimal w^* can be found
- Now optimal b^* can be obtained as

$$b^* = -\frac{1}{2}(max_{i:y_i=-1}\ (w^*)^T x_i + min_{i:y_i=+1}\ (w^*)^T x_i)$$

- **Validation check of a known data**: For a given x the model can calculate

$$w^T x + b = \sum_{i=1}^{n} (\alpha_i y_i x_i)^T x + b$$
$$= \sum_{i=1}^{n} \alpha_i y_i \langle x_i^T, x \rangle + b$$

Steps of the algorithm

1. Read the input data and structure it in a tabular format.
2. Normalize the raw data.
3. Extract the features.
4. Find the optimal penalty parameter and degree of the fitting SVM Polynomial using cross validated grid search.
5. Define a support vector classifier with the obtained optimal parameters.
6. Fit it on the training data and compute the predictions for the test data.

Observations and Comments

- We have used Jupyter notebook for implementing the algorithm.

Observations and Comments

- We have used Jupyter notebook for implementing the algorithm.
- Advantages:

Observations and Comments

- We have used Jupyter notebook for implementing the algorithm.
- Advantages:
  - The SVM based algorithm is able to achieve significantly high accuracy on the test data indicating its robustness.

Observations and Comments

- We have used Jupyter notebook for implementing the algorithm.
- Advantages:
  - The SVM based algorithm is able to achieve significantly high accuracy on the test data indicating its robustness.
  - It is computationally cheap.

Observations and Comments

- We have used Jupyter notebook for implementing the algorithm.
- Advantages:
  - The SVM based algorithm is able to achieve significantly high accuracy on the test data indicating its robustness.
  - It is computationally cheap.
  - The algorithm is also able to efficiently predict classes for non linear data or for data in high dimensions using RBF based kernels.

Observations and Comments

- We have used Jupyter notebook for implementing the algorithm.
- Advantages:
  - The SVM based algorithm is able to achieve significantly high accuracy on the test data indicating its robustness.
  - It is computationally cheap.
  - The algorithm is also able to efficiently predict classes for non linear data or for data in high dimensions using RBF based kernels.
- Limitations:

Observations and Comments

- We have used Jupyter notebook for implementing the algorithm.
- Advantages:
  - The SVM based algorithm is able to achieve significantly high accuracy on the test data indicating its robustness.
  - It is computationally cheap.
  - The algorithm is also able to efficiently predict classes for non linear data or for data in high dimensions using RBF based kernels.
- Limitations:
  - This algorithm does not work well when number of data points is significantly less than the number of features.

Observations and Comments

- We have used Jupyter notebook for implementing the algorithm.
- Advantages:
  - The SVM based algorithm is able to achieve significantly high accuracy on the test data indicating its robustness.
  - It is computationally cheap.
  - The algorithm is also able to efficiently predict classes for non linear data or for data in high dimensions using RBF based kernels.
- Limitations:
  - This algorithm does not work well when number of data points is significantly less than the number of features.
  - It performs poorly when the data is not separable.

Observations and Comments

- We have used Jupyter notebook for implementing the algorithm.
- Advantages:
  - The SVM based algorithm is able to achieve significantly high accuracy on the test data indicating its robustness.
  - It is computationally cheap.
  - The algorithm is also able to efficiently predict classes for non linear data or for data in high dimensions using RBF based kernels.
- Limitations:
  - This algorithm does not work well when number of data points is significantly less than the number of features.
  - It performs poorly when the data is not separable.
- Overall, the time complexity is $O(\max(n, d) * (\min(n, d)^2)$.

References

- Time Complexity Analysis of SVM
- Cross Validation and Grid Search
- Using SVMs
- Support Vector Machines
- Chauhan, Vinod Kumar, Kalpana Dahiya, and Anuj Sharma. "Problem formulations and solvers in linear SVM: a review." Artificial Intelligence Review 52.2 (2019): 803-855.
- Wind Speed Prediction