Chapter 1
Introduction to SVM

Hadi Veisi

Abstract In this chapter, a review of the machine learning (ML) and pattern recognition concepts is given, and basic ML techniques (supervised, unsupervised, and reinforcement learning) are described. Also, a brief history of ML development from the primary works before the 1950s (including Bayesian theory) up to the most recent approaches (including deep learning) is presented. Then, an introduction to the support vector machine (SVM) with a geometric interpretation is given, and its basic concepts and formulations are described. A history of SVM progress (from Vapnik's primary works in the 1960s up to now) is also reviewed. Finally, various ML applications of SVM in several fields such as medical, text classification, and image classification are presented.

Keywords Machine leaning · Pattern recognition · Support vector machine · History

1.1 What Is Machine Learning?

Recognizing a person from his/her face, reading a handwritten letter, understanding a speech lecture, deciding to buy the stock of a company after analyzing the company's profile, and driving a car in a busy street are some examples of using human intelligence. Artificial Intelligence (AI) refers to the simulation of human intelligence in machines, i.e., computers. Machine Learning (ML) as a subset of AI is the science of automatically learning computers from experiences to do intelligent and human-like tasks. Similar to other actions in computer science and engineering, ML is realized by computer algorithms that can learn from their environment (i.e., data) and can generalize this training to act intelligently in new environments. Nowadays, computers can recognize people from their face using face recognition algorithms, converting a handwritten letter to its editable form using handwritten recognition,

H. Veisi (✉)
Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran
e-mail: h.veisi@ut.ac.ir

J. A. Rad et al. (eds.), *Learning with Fractional Orthogonal Kernel Classifiers in Support Vector Machines*, Industrial and Applied Mathematics,
https://doi.org/10.1007/978-981-19-6553-1_1

understand a speech lecture using speech recognition and natural language understanding, buy a stock of a company using algorithmic trading methods, and can drive a car automatically in self-driving cars. The term machine learning was coined by Arthur Samuel in 1959, an American pioneer in the field of computer gaming and artificial intelligence that defines this term as “it gives computers the ability to learn without being explicitly programmed” Samuel (2000). In 1997, Tom Mitchell, an American computer scientist and a former Chair of the Machine Learning Department at the Carnegie Mellon University (CMU) gave a mathematical and relational definition that “A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E” Mitchell (1997). So, if you want your ML program to predict the growth of a stock (task T) to decide for buying it, you can run a machine learning algorithm with data about past price patterns of this stock (experience E, which is called training data) and, if it has successfully “learned”, it will then do better at predicting future price (performance measure P). The primary works in ML return to the 1950s and this field has received several improvements during the last 70 years. There is a short history of ML:

- **Before the 1950s**: Several ML-related theories have been developed including Bayesian theory Bayes (1991), Markov chain Gagniuc (2017), regression, and estimation theories Fisher (1922). Also, Donald Hebb in 1949 Hebb (1949) presented his model of brain neuron interactions which is the basis of McCulloch-Pitts’s neural networks McCulloch and Pitts (1943).
- **The 1950s**: In this decade, ML pioneers have proposed the primary ideas and algorithms for machine learning. The Turing test, originally called the imitation game, was proposed by Alan Turing as a test of a machine’s ability to exhibit intelligent behavior equivalent to, or indistinguishable from, that of a human. Arthur Samuel of IBM developed a computer program for playing checkers Samuel (1959). Frank Rosenblatt extended Hebb’s learning model of brain cell interaction with Arthur Samuel’s ideas and created the perceptron Rosenblatt (1958).
- **The 1960s**: Bayesian methods are introduced for probabilistic inference Solomonoff (1964). The primary idea of Support Vector Machines (SVMs) is given by Vapnik and Lerner Vapnik (1963). Widrow and Hoff developed the delta learning rules for neural networks which was the precursor of the backpropagation algorithm Vapnik (1963). Sebestyen Sebestyen (1962) and Nilsson Nilsson (1965) proposed the nearest neighbor idea. Donald Michie used reinforcement learning to play Tic-tac-toe Michie (1963). The decision tree was introduced by Morgan and Sonquist Morgan and Sonquist (1963).
- **The 1970s**: The quit years which is also known as AI Winter caused by pessimism about machine learning effectiveness due to the limitation of the ML methods in solving only linearly separable problems Minsky and Papert (1969).
- **The 1980s**: The birth of brilliant ideas resulted in renewed enthusiasm. Backpropagation publicized by Rumelhart et al. (1986) causes a resurgence in machine learning. Hopfield popularizes his recurrent neural networks Hopfield (Hopfield). Watkins develops Q-learning in reinforcement learning Watkins (1989).

Fukushima published his work on the neocognitron neural network Fukushima (1988) which later inspires Convolutional Neural Networks (CNNs). Boltzmann machine Hinton (1983) was proposed which was later used in Deep Belief Networks (DBNs).

- **The 1990s**: This is the decade for the birth of today's form of SVM by Vapnik and his colleagues in Boser et al. (1992) and is extended in Vapnik (1995) and Cortes and Vapnik (1995). In these years, ML works shift from knowledge-driven and rule-based approaches to the data-driven approach, and other learning algorithms such as Recurrent Neural Networks (RNNs) are introduced. Hochreiter and Schmidhuber invent long short-term memory (LSTM) recurrent neural networks Hochreiter and Schmidhuber (1997) which became a practicality successful method for sequential data modeling. IBM's Deep Blue beats the world champion at chess, the grand master Garry Kasparov Warwick (2017). Tin Kam Ho introduced random decision forests Ho (1995). Boosting algorithms are proposed Schapire (1990).
- **The 2000s**: Using ML methods in real applications, dataset creation, and organizing ML challenges become widespread. Support Vector Clustering and other Kernel methods were introduced Ben-Hur et al. (2001). A deep belief network was proposed by Hinton which is among the starting points for deep learning Hinton et al. (2006).
- **The 2010s**: Deep learning becomes popular and has overcome most ML methods, results in becoming integral to many real-world applications Goodfellow et al. (2016). Various deep neural networks such as autoencoders Liu et al. (2017), Convolutional Neural Networks (CNNs) Khan et al. (2020), and Generative Adversarial Networks (GANs) Pan et al. (2019) were introduced. ML achieved higher performance than human in various fields such as lipreading (e.g., LipNet Assael et al. (2016)), playing Go (e.g., Google's AlphaGo and AlphaGo Zero programs Silver et al. (2017)), and information retrieval using natural language processing (e.g., IBM's Watson in Jeopardy competition Ferrucci et al. (2013)).

The highly complex nature of most real-world problems often means that inventing specialized algorithms that will solve them perfectly every time is impractical, if not impossible. Examples of machine learning problems include “Can we recognize the spoken words by only looking at the lip movements?”, “Is this cancer in this mammogram?”, “Which of these people are good friends with each other?”, and “Will this person like this movie?”. Such problems are excellent targets for ML, and, in fact, machine learning has been applied to such problems with great success, as mentioned in the history. Machine learning is also highly related to another similar topic, pattern recognition. These two terms can now be viewed as two facets of the same fields; however, machine learning grew out of computer science whereas pattern recognition has its origins in engineering Bishop (2006). Another similar topic is data mining which utilizes ML techniques in discovering patterns in large data and transforming row data into information and decision. Within the field of data analytics, machine learning is used to devise complex models and algorithms that lend themselves to prediction which is known as predictive analytics in commercial applications. These analytical models allow researchers, data scientists, engineers, and analysts to "produce reliable, repeatable decisions and results" and uncover "hidden insights" through learning from historical relationships and trends in the data (i.e., input).

1.1.1 Classification of Machine Learning Techniques

Machine learning techniques are classified into three following categories, depending on the nature of the learning data or learning process:

1. **Supervised learning:** In this type of learning, there is a supervisor to teach machines in learning a concept. This means that the algorithm learns from labeled data (i.e., training data) which include example data and related target responses, i.e., input and output pairs. If we assume the ML algorithm as a system (e.g., a face identification), in the training phase of the system, we provide both input sample (e.g., an image from a face) and the corresponding output (e.g., the ID of the person to whom face belongs) to the system. The collection of labeled data requires skilled human agents (e.g., a translator to translate a text from a language to another) or a physical experiment (e.g., determining whether there is rock or metal near to a sonar system of a submarine) that is costly and time-consuming. The supervised learning methods can be divided into classification and regression. When the number of classes of the data is limited (i.e., the output label of the data is a discrete variable) the learning is called classification (e.g., classifying an email to spam and not-spam classes), and when the output label is a continuous variable (e.g., the price of a stock index) the topic is called regression. Examples of the most widely used supervised learning algorithms are SVM Boser et al. (1992), Vapnik (1995), Cortes and Vapnik (1995), artificial neural networks (e.g., multi-layer perceptron Rumelhart et al. (1986), LSTM Hochreiter and Schmidhuber (1997), Gers et al. (1999)), linear and logistic regression Cramer (2002), Naïve Bayes Hand and Yu (2001), decision trees Morgan and Sonquist (1963), and K-Nearest Neighbor (KNN) Sebestyen (1962), Nilsson (1965).
2. **Unsupervised learning**: In this case, there is not any supervision in the learning and the ML algorithm works on the unlabeled data. It means that the algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns based on the similarities in the data. This type of algorithm tends to restructure the data and cluster them. From a system viewpoint, this kind of learning receives sample data as the input (e.g., the human faces) without the corresponding output and groups with similar samples in the same clusters. The categorization of unlabeled data is commonly called clustering. Also, association as another type of unsupervised learning refers to methods which can discover rules that describe large portions of data, such as people who buy product X also tend to buy the other product Y. Dimensionality

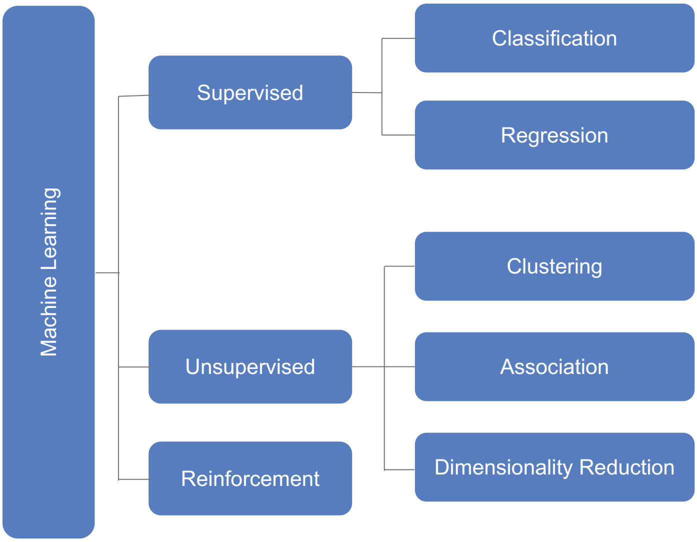


Fig. 1.1 Various types of machine learning

reduction as another type of unsupervised learning denotes the methods transform data from a high-dimensional space into a low-dimensional space. Examples of well-known unsupervised learning methods Xu and Wunsch (2005) are k-means, hierarchical clustering, density-based spatial clustering of applications with noise (DBSCAN), neural networks (e.g., autoencoders, self-organizing map), Expectation-Maximization (EM), and Principal Component Analysis (PCA).

3. **Reinforcement learning**: In this category of learning, an agent can learn from an action-reward mechanism by interacting with an environment just like the learning process of a human to play chess game by exercising and training by trial and error. Reinforcement algorithms are presented with examples and without labels but receiving positive (e.g., reward) or negative (e.g., penalty) feedback from the environment. Two widely used reinforcement models are Markov Decision Process (MDP) and Q-learning Sutton and Barto (2018).

A summary of machine learning types is summarized in Fig. 1.1. In addition to the mentioned three categories, there is another type called semi-supervised learning (that is also referred to as either transductive learning or inductive learning) which falls between supervised and unsupervised learning methods. It combines a small amount of labeled data with a large amount of unlabeled data for training.

From another point of view, machine learning techniques are classified into the generative approach and discriminative approach. Generative models explicitly model the actual distribution of each class of data while discriminative models learn the (hard or soft) boundary between classes. From the statistical viewpoint, both of these approaches finally predict the conditional probability $P(Class|Data)$ but both models learn different probabilities. In generative methods, joint distribution $P(Class, Data)$ is learned and the prediction is performed according to this distribution. On the other side, discriminative models do predictions by estimating conditional probability $P(Class|Data)$.

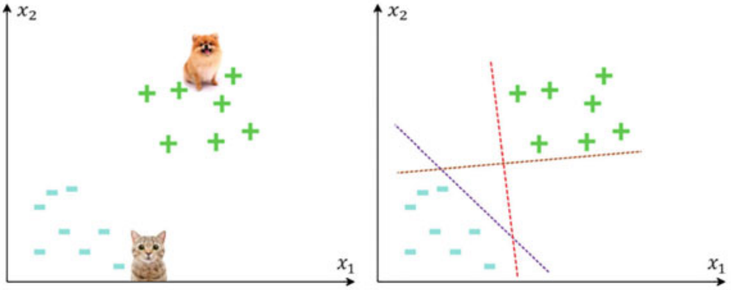
Examples of generative methods are deep generative models (DGMs) such as Variational Autoencoder (VAE) and GANs, Naïve Bayes, Markov random fields, and Hidden Markov Models (HMM). SVM is a discriminative method that learns the decision boundary like some other methods such as logistic regression, traditional neural networks such as multi-layer perceptron (MLP), KNN, and Conditional Random Fields (CRFs).

1.2 What Is the Pattern?

In ML, we seek to design and build machines that can learn and recognize patterns, as is also called pattern recognition. To do this, the data need to have regularity or arrangement, called pattern, to be learned by ML algorithms. The data may be created by humans such as stock price or a signature or have a natural nature such as speech signals or DNS. Therefore, a pattern includes elements that are repeated in a predictable manner. The patterns in natural data, e.g., speech signals, are often chaotic and stochastic and do not exactly repeat. There are various types of natural patterns which include spirals, meanders, waves, foams, tilings, cracks, and those created by symmetries of rotation and reflection. Some types of patterns such as a geometric pattern in an image can be directly observed while abstract patterns in a huge amount of data or a language may become observable after analyzing the data using pattern discovery methods. In both cases, the underlying mathematical structure of a pattern can be analyzed by machine learning techniques which are mainly empowered by mathematical tools. The techniques can learn the patterns to predict or recognize them or can search them to find the regularities. Accordingly, if a dataset suffers from any regularities and repeatable templates, the modeling result of ML techniques will not be promising.

1.3 An Introduction to SVM with a Geometric Interpretation

Support Vector Machine (SVM), also known as support vector network, is a supervised learning approach used for classification and regression. Given a set of training labeled examples belonging to two classes, the SVM training algorithm builds a decision boundary between the samples of these classes. SVM does this in such a way that optimally discriminates between two classes by maximizing the margin between two data categories. For data samples in an N-dimensional space, SVM constructs an $N - 1$-dimensional separating hyperplane to discriminate two classes. To describe



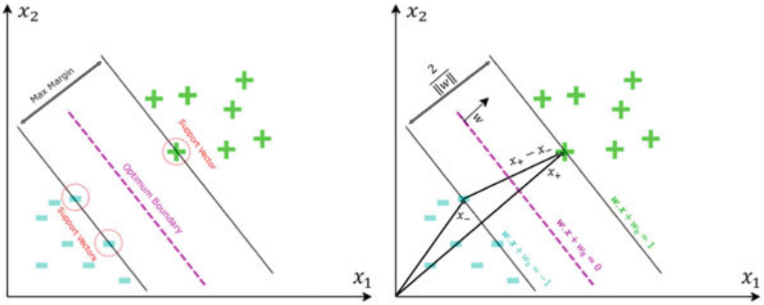
(a) Samples of data in the vector space. (b) Several possible separating lines.

Fig. 1.2 **a** A binary classification example, **b** Some possible decision boundaries

the SVM, assume a binary classification example in a two-dimensional space, e.g., distinguishing a cat from a dog using the values of their height (x_1) and weight (x_2). An example of the training labeled samples for these classes in the feature space is given in Fig. 1.2a, in which the samples for the dog are considered as the positive samples and the cat samples are presented as the negative ones. To do the classification, probably the primary intuitive approach is to draw a separative line between the positive and negative samples. However, as it is shown in Fig. 1.2b, there are many possible lines to be the decision boundary between these classes. Now, the question is which line is better and should be chosen as the boundary?

Although all the lines given in Fig. 1.2b are a true answer to do the classification, neither of them seems the best fit. Alternatively, a line that is drawn between the two classes which have the maximum distance from both classes is the better choice. To do this, the data points that lie closest to the decision boundary are the most difficult samples to classify and they have a direct bearing on the optimum location of the boundary. These samples are called support vectors that are closer to the decision boundary and influence the position and orientation (see Fig. 1.3). According to these vectors, the maximum distance between the two classes is determined. This distance is called margin and a decision line that is half this margin seems to be the optimum boundary. This line is such that the margin is maximized which is called the maximum-margin line between the two classes. SVM classifier attempts to find this optimal line.

In SVM, to construct the optimal decision boundary, a vector **w** is considered to be perpendicular to the margin. Now, to classifying an unknown vector **x**, we can project it onto w by computing $\mathbf{w}.\mathbf{x}$ and determine on which side of the decision boundary **x** lies by calculating $\mathbf{w}.\mathbf{x} \geq t$ for a constant threshold t. It means that if the values of $\mathbf{w}.\mathbf{x}$ are more than t, i.e., it is far away, sample **x** is classified as a positive example. By assuming $t = -w_0$, the given decision rule can be given as $\mathbf{w}.\mathbf{x} + w_0 \geq 0$. Now, the question is, how we can determine the values of **w** and w_0? To do this, the following constraints are considered which means a sample is classified as positive if the value is equal or greater than 1, it is classified as negative if the value of -1 or less:



(a) Support vectors and optimum decision boundary.

(b) Decision boundary equation and maximum margin.

Fig. 1.3 **a** SVM optimum decision boundary, **b** Definition of the notations

$$\mathbf{w}.\mathbf{x}_+ + w_0 \geq 1, \quad and \quad \mathbf{w}.\mathbf{x}_- + w_0 \leq -1. \tag{1.1}$$

These two equations can be integrated into an inequality, as in Eq. 1.2 by introducing a supplementary variable, y_i which is equal to $+1$ for positive samples and is equal to -1 for negative samples. This inequality is considered as equality, i.e., $y_i(\mathbf{w}.\mathbf{x}_i + w_0) - 1 = 0$, to define the main constraint of the problem which means the examples lying on the margins (i.e., *support vectors*) to be constrained to 0. This equation is equivalence to a line that is the answer to our problem. This decision boundary line in this example becomes a hyperplane in the general N-dimensional case.

$$y_i(\mathbf{w}.\mathbf{x}_i + w_0) - 1 \geq 0. \tag{1.2}$$

To find the maximum margin that separating positive and negative examples, we need to know the width of the margin. To calculate the width of the margin, $(\mathbf{x}_+ - \mathbf{x}_-)$ need to be projected onto unit normalized $\frac{\mathbf{w}}{\|\mathbf{w}\|}$. Therefore, the width is computed as $(\mathbf{x}_+ - \mathbf{x}_-).\frac{\mathbf{w}}{\|\mathbf{w}\|}$ in which by using $y_i(\mathbf{w}.\mathbf{x}_i + w_0) - 1 = 0$ to substituting $\mathbf{w}.\mathbf{x}+ = 1 - w_0$ and $\mathbf{w}.\mathbf{x}- = 1 + w_0$ in that, the final value for width is obtained as $2\|\mathbf{w}\|$ (see Fig. 1.3). Finally, maximizing this margin is equivalent to Eq. 1.3 which is a quadratic function:

$$\min_{\mathbf{w},w_0} \frac{1}{2}\|\mathbf{w}\|^2,$$
$$y_i(\mathbf{w}.\mathbf{x}_i + w_0) - 1 \geq 0. \tag{1.3}$$

This is a constrained optimization problem and can be solved by the Lagrange multiplier method. After writing the Lagrangian equation as in Eq. 1.4, and computing the partial derivative with respect to $\mathbf{w}$ and setting it to zero results in $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$ and with respect to w_0 and setting it to zero gives $\sum_i \alpha_i y_i$. It means that $\mathbf{w}$ is a linear combination of the samples. Using these values in Eq. 1.4 results in a Lagrangian

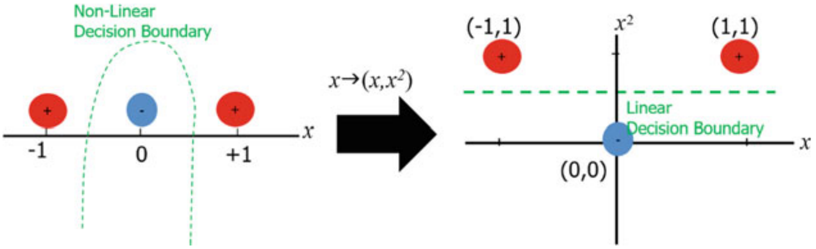


Fig. 1.4 Transforming data from a nonlinear space into a linear higher dimensional space

equation in which the problem depends only on dot products of pairs of data samples. Also, $\alpha_i = 0$ for the training examples that are not support vectors that means these examples do not affect the decision boundary. Another interesting fact about this optimization problem is that it is a convex problem and it is guaranteed to always find a global optimum.

$$L(\mathbf{w}, w_0) = \frac{1}{2}\|\mathbf{w}\|^2 + \sum_i [y_i(\mathbf{w}.\mathbf{x}_i + w_0) - 1]\alpha_i. \tag{1.4}$$

The above-described classification problem and its solution using the SVM assumes the data is linearly separable. However, in most real-life applications, this assumption is not correct and most problems are not classified simply using a linear boundary. The SVM decision boundary is originally linear Vapnik (1963) but has been extended to handle nonlinear cases as well Boser et al. (1992). To do this, SVM proposes a method called kernel trick in which an input vector is transformed using a nonlinear function like $\phi(.)$ into a higher dimensional space. Then, in this new space, the maximum-margin linear boundary is found. It means that a nonlinear problem is converted into a linearly separable problem in the new higher dimensional space without affecting the convexity of the problem. A simple example of this technique is given in Fig. 1.4 in which one-dimensional data samples, x_i, are transformed into two-dimensional space using $(x_i, x_i \times x_i)$ transform. In this case, the dot product of two samples, i.e., $x_i.x_j$, in the optimization problem is replaced with $\phi(x_i).\phi(x_j)$. In practice, if we have a function like such that $K(x_i, x_j) = \phi(x_i).\phi(x_j)$, then we do not need to know the transformation $\phi(.)$ and only function $K(.)$ (which is called the *kernel function*) is required. Some common kernel functions are linear, polynomial, sigmoid, and radial basis functions (RBF).

Although the kernel trick is a clever method to handle the nonlinearity, the SVM still assumes that the data is linearly separable in this transformed space. This assumption is not true in most real-world applications. Therefore, another type of SVM is proposed which is called *soft-margin* SVM Cortes and Vapnik (1995). The described SVM method up to now is known as *hard-margin* SVM. As given, hard-margin SVMs assume the data is linearly separable without any errors, whereas soft-margin SVMs allow some misclassification and results in a more robust decision in nonlinearly separable data. Today, soft-margin SVMs are the most common SVM techniques in ML which utilize so-called slack variables in the optimization problem to control the amount of misclassification.

1.4 History of SVMs

Vladimir Vapnik, the Russian statistician, is the main originator of the SVM technique. The primary work on the SVM algorithm was proposed by Vapnik (1963) as the Generalized Portrait algorithm for pattern recognition. However, it was not the first algorithm for pattern recognition, and Fisher in 1936 had proposed a method for this purpose Fisher (1936). Also, Frank Rosenblatt had been proposed the perceptron linear classifier which was an early feedforward neural network Rosenblatt (1958). One year after the primary work of Vapnik and Lerner, in 1964, Vapnik further developed the Generalized Portrait algorithm Vapnik and Chervonenkis (1964). In this year, a geometrical interpretation of the kernels was introduced in Aizerman et al. (1964) as inner products in a feature space. The kernel theory which is the main concept in the development of SVM and is called "kernel trick" was previously proposed in Aronszajn (1950). In 1965, a large margin hyperplane in the input space was introduced in Cover (1965) which is another key idea of the SVM algorithm. At the same time, a similar optimization concept was used in pattern recognition by Mangasarian (1965). Another important research that defines the basic idea of the soft-margin concept in SVM was introduced by Smith (1968). This idea was presented as the use of slack variables to overcome the problem of noisy samples that are not linearly separable. In the history of SVM development, the breakthrough work is the formulation of statistical learning framework or VC theory proposed by Vapnik and Chervonenkis (1974) which presents one of the most robust prediction methods. It is not surprising to say that the rising of SVM was in this decade and this reference has been translated from Russian into other languages such as German Vapnik and Chervonenkis (1979) and English Vapnik (1982). The use of polynomial kernel in SVM was proposed by Poggio (1975) and the improvement of kernel techniques for regression was presented by Wahba (1990). Studying the connection between neural networks and kernel regression was done by Poggio and Girosi (1990). The improvement of the previous work on slack variables in Smith (1968) was done by Bennett and Mangasarian (1992). Another main milestone in the development of SVM is in 1992 in which SVM has been presented in its today's form by Boser et al. (1992). In this work, the optimal margin classifier of linear classifiers (from Vapnik (1963)) was extended to nonlinear cases by utilizing the kernel trick to maximum-margin hyperplanes Aizerman et al. (1964). In 1995, soft margin of SVM classifiers to handle noisy and not linearly separable data was introduced using slack variables by Cortes and Vapnik (1995). In 1996, the algorithm was extended to the case of regression Drucker et al. (1996) which is called Support Vector Regression (SVR). The rapid growth of SVM and using this technique in various applications has been increased after 1995. Also, the theoretical aspects of SVM have been

Table 1.1 A brief history of SVM development

Decade	Year	Researcher(s)	SVM development
1950	1950	Aronszajn (1950)	Introducing the “Theory of Reproducing Kernels”
1960	1963	Vapnik and Lerner (1963)	Introducing the Generalized Portrait algorithm (the algorithm implemented by support vector machines is a nonlinear generalization of the Generalized Portrait algorithm)
	1964	Vapnik (1964)	Developing the Generalized Portrait algorithm
	1964	Aizerman et al. (1964)	Introducing the geometrical interpretation of the kernels as inner products in a feature space
	1965	Cover (1965)	Discussing large margin hyperplanes in the input space and also sparseness
	1965	Mangasarian (1965)	Studding optimization techniques for pattern recognition similar to large margin hyperplanes
	1968	Smith (1968)	Introducing the use of slack variables to overcome the problem of noise and non-separability
1970	1973	Duda and Hart (1973)	Discussing large margin hyperplanes in the input space
	1974	Vapnik and Chervonenkis (1974)	Writing a book on “statistical learning theory” (in Russian) which can be viewed as the starting of SVMs
	1975	Poggio (1975)	Proposing the use of polynomial kernel in SVM
	1979	Vapnik and Chervonenkis (1979)	Translating of Vapnik and Chervonenkis’s 1974 book to German
1980	1982	Vapnik (1982)	Writing an English translation of his 1979 book
1990	1990	Poggio and Girosi (1990)	Studying the connection between neural networks and kernel regression
	1990	Wahba (1990)	Improving the kernel method for regressing
	1992	Bennett and Mangasarian (1992)	Improving Smith’s 1968 work on slack variables
	1992	Boser et al. (1992)	Presenting SVM in today’s form at the COLT 1992 conference
	1995	Cortes and Vapnik (1995)	Introducing the soft-margin classifier
	1996	Drucker et al. (1996)	Extending the algorithm to the case of regression, called SVR
	1997	Muller et al. (1997)	Extending SVM for time-series prediction
	1998	Bartlett (1998)	Providing the statistical bounds on the generalization of hard margin
2000	2000	Shawe-Taylor and Cristianini (2000)	Giving the statistical bounds on the generalization of soft margin and the regression case
	2001	Ben-Hur et al. (2001)	Extending SVM to the unsupervised case
	2005	Duan and Keerthi (2005)	Extending SVM from the binary classification into multiclass SVM
2010	2011	Polson and Scott (2011)	Studding graphical model representation of SVM and its Bayesian interpretation
	2017	Wenzel et al. (2017)	Developing a scalable version of Bayesian SVM for big data applications

studied and it has been extended in other domains than the classification. The statistical bounds on the generalization of hard margin were given by Bartlett (1998) and it was presented for soft margin and the regression case in 2000 by Shawe-Taylor and Cristianini (2000). The SVM was originally developed for supervised learning which has been extended to the unsupervised case in Ben-Hur et al. (2001) called support vector clustering. Another improvement of SVM was its extension from the binary classification into multiclass SVM by Duan and Keerthi in Duan and Keerthi (2005) by distinguishing between one of the labels and the rest (one-versus-all) or between every pair of classes (one-versus-one). In 2011, SVM was analyzed as a graphical model and it was shown that it admits a Bayesian interpretation using data augmentation technique Polson and Scott (2011). Accordingly, a scalable version of the Bayesian SVM was developed in Wenzel et al. (2017) enabling the application of Bayesian SVMs in big data applications. A summary of the related researches to SVM development is given in Table 1.1.

1.5 SVM Applications

Today, machine learning algorithms are on the rise and are widely used in real applications. Every year new techniques are proposed that overcome the current leading algorithms. Some of them are only little advances or combinations of existing algorithms and others are newly created and lead to astonishing progress. Although deep learning techniques are dominant in many real applications such as image processing (e.g., for image classification) and sequential data modeling (e.g., in natural language processing tasks such as machine translation), these techniques require a huge amount of training data for success modeling. Large-scale labeled datasets are not available in many applications in which other ML techniques (called classical ML methods) such as SVM, decision tree, and Bayesian family methods have higher performance than deep learning techniques. Furthermore, there is another fact in ML. Each task in ML applications can be solved using various methods; however, there is no single algorithm that will work well for all tasks. This fact is known as the No Free Lunch Theorem in ML Wolpert (1996). Each task that we want to solve has its idiosyncrasies and there are various ML algorithms to suit the problem. Among the non-deep learning methods, SVM, as a well-known machine learning technique, is widely used in various classification and regression tasks today due to its high performance and reliability across a wide variety of problem domains and datasets Cervantes et al. (2020). Generally, SVM can be applied to any ML task in any application such as computer vision and image processing, natural language processing (NLP), medical applications, biometrics, and cognitive science. In the following, some common applications of SVM are reviewed.

- **ML Applications in Medical**: SVM is widely applied in medical applications including medical image processing (i.e., a cancer diagnosis in mammography Azar and El-Said (2014)), bioinformatics (i.e., patients and gene classification) Byvatov and Schneider (2003), and health signal analysis (i.e., electrocardiogram signal classification for detection for identifying heart anomalies Melgani and Bazi (2008) and electroencephalogram signal processing in psychology and neuroscience Li et al. (2013)). SVM which is among the successful methods in this field is used in diagnosis and prognosis of various types of diseases. Also, SVM is utilized for identifying the classification of genes, patients based on genes, and other biological problems Pavlidis et al. (2001).
- **Text Classification**: There are various applications for text classification including topic identification (i.e., categorization of a given document into predefined classes such as scientific, sport, or political); author identification/verification of a written document, spam, and fake news/email/comment detection; language identification (i.e., determining the language of a document); polarity detection (e.g., finding that a given comment in a social media or e-commerce website is positive or negative); and word sense disambiguation (i.e., determining the meaning of a disembogues word in a sentence such as “bank”). SVM is a competitive method for the other ML techniques in this area Joachims (1999), Aggarwal and Zhai (2012).
- **Image Classification**: The process of assigning a label to an image is generally known as image recognition which can be used in various fields such as in biometrics (e.g., face detection/identification/verification), medical (e.g., processing MRI and CT images for disease diagnosis), object recognition, remote sensing classification (e.g., categorization of satellite images), automated image organization for social networks and websites, visual searches, and image retrieval. Although deep learning techniques, specially CNNs are leading this area for the representation and feature extraction, SVM is utilized commonly as the classifier Miranda et al. (2016), Tuia et al. (2011), both with classical image processing-based techniques and deep neural network methods Li (2019). The image recognition services are now generally offered by the AI teams of technology companies such as Microsoft, IBM, Amazon, and Google.
- **Steganography Detection**: Steganography is the practice of concealing a message in an appropriate multimedia carrier such as an image, an audio file, or a video file. This technique is used for secure communication in security-based organizations to concealing both the fact that a secret message is being sent and its contents. This method has the advantage over cryptography in which only the content of a message is protected. On the other hand, the act of detecting whether an image (or other files) is stego or not is called steganalysis. The analysis of an image to determining if it is a stego image or not is a binary classification problem in which SVM is widely used Li (2011).

References

Aggarwal, C.C., Zhai, C.: A survey of text classification algorithms. In: Aggarwal, C.C., Zhai, C. (eds.) Mining Text Data, 163–222. Springer, Boston (2012)

Aizerman, M.A., Braverman, E.M., Rozonoer, L.I.: Theoretical foundations of the potential function method in pattern recognition learning. Autom. Remote. **25**, 821–837 (1964)

Aronszajn, N.: Theory of reproducing kernels. Trans. Am. Math. Soc. **68**, 337–404 (1950)

Assael, Y.M., Shillingford, B., Whiteson, S., De Freitas, N.: Lipnet: End-to-end Sentence-Level Lipreading (2016). arXiv:1611.01599

Azar, A.T., El-Said, S.A.: Performance analysis of support vector machines classifiers in breast cancer mammography recognition. Neural. Comput. Appl. **24**, 1163–1177 (2014)

Bartlett, P.L.: The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. IEEE Trans. Inf. Theory **44**, 525–536 (1998)

Bayes, T.: An essay towards solving a problem in the doctrine of chances. MD Comput. **8**, 157 (1991)

Ben-Hur, A., Horn, D., Siegelmann, H.T., Vapnik, V.: Support vector clustering. J. Mach. Learn. Res. **2**, 125–137 (2001)

Bennett, K.P., Mangasarian, O.L.: Robust linear programming discrimination of two linearly inseparable sets. Optim. Methods. Softw. **1**, 23–34 (1992)

Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Singapore (2006)

Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: COLT92: 5th Annual Workshop Computers Learning Theory, PA (1992)

Byvatov, E., Schneider, G.: Support vector machine applications in bioinformatics. Appl. Bioinf. **2**, 67–77 (2003)

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., Lopez, A.: A comprehensive survey on support vector machine classification: applications, challenges and trends. Neurocomputing **408**, 189–215 (2020)

Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**, 273–297 (1995)

Cover, T.M.: Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE Trans. Elect. Comput. **3**, 326–334 (1965)

Cramer, J.S.: The origins of logistic regression (Technical report). Tinbergen Inst. 167–178 (2002)

Drucker, H., Burges, C.J., Kaufman, L., Smola, A., Vapnik, V.: Support vector regression machines. Adv. Neural Inf. Process Syst. **9**, 155–161 (1996)

Duan, K. B., Keerthi, S. S.: Which is the best multiclass SVM method? An empirical study. In: International Workshop on Multiple Classifier Systems, pp. 278–285. Springer, Heidelberg (2005)

Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. Wiley, New York (1973)

Ferrucci, D., Levas, A., Bagchi, S., Gondek, D., Mueller, E.T.: Watson: beyond jeopardy! Artif. Intell. **199**, 93–105 (2013)

Fisher, R.A.: The goodness of fit of regression formulae, and the distribution of regression coefficients. J. R. Stat. Soc. **85**, 597–612 (1922)

Fisher, R.A.: The use of multiple measurements in taxonomic problems. Ann. Eugen. **7**, 179–188 (1936)

Fukushima, K.: Neocognitron: a hierarchical neural network capable of visual pattern recognition. Neural Netw. **1**, 119–130 (1988)

Gagniuc, P.A.: Markov Chains: from Theory to Implementation and Experimentation. Wiley, Hoboken, NJ (2017)

Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: continual prediction with LSTM. In: International Conference on Artificial Neural Networks, Edinburgh (1999)

Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: Deep Learning. MIT Press, England (2016)

Hand, D.J., Yu, K.: Idiot's Bayes-not so stupid after all? Int. Stat. Rev. **69**, 385–398 (2001)

Hebb, D.: The Organization of Behavior. Wiley, New York (1949)

Hinton, G.E.: Analyzing Cooperative Computation. 5th COGSCI, Rochester (1983)

Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**, 1527–1554 (2006)

Ho, T.K.: Random decision forests. In: Proceedings of 3rd International Conference. on Document Analysis and Recognition, Montreal (1995)

Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**, 1735–1780 (1997)

Hopfield, J.J.: Neurons with graded response have collective computational properties like those of two-state neurons. Proc. Natl. Acad. Sci. **81**, 3088–3092 (1984)

Joachims, T.: Transductive inference for text classification using support vector machines. In: ICML 99: Proceedings of the Sixteenth International Conference on Machine Learning, 200–209 (1999)

Khan, A., Sohail, A., Zahoora, U., Qureshi, A.S.: A survey of the recent architectures of deep convolutional neural networks. Artif. Intell. Rev. **53**, 5455–5516 (2020)

Li, B., He, J., Huang, J., Shi, Y.Q.: A survey on image steganography and steganalysis. J. Inf. Hiding Multimedia Signal Process. **2**, 142–172 (2011)

Li, S., Zhou, W., Yuan, Q., Geng, S., Cai, D.: Feature extraction and recognition of ictal EEG using EMD and SVM. Comput. Biol. Med. **43**, 807–816 (2013)

Li, Y., Li, J., Pan, J.S.: Hyperspectral image recognition using SVM combined deep learning. J. Internet Technol. **20**, 851–859 (2019)

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., Alsaadi, F.E.: A survey of deep neural network architectures and their applications. Neurocomputing **234**, 11–26 (2017)

Mangasarian, O.L.: Linear and nonlinear separation of patterns by linear programming. Oper. Res. **13**, 444–452 (1965)

McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biol. **5**, 115–133 (1943)

Melgani, F., Bazi, Y.: Classification of electrocardiogram signals with support vector machines and particle swarm optimization. IEEE Trans. Inf. Technol. Biomed. **12**, 667–677 (2008)

Michie, D.: Experiments on the mechanization of game-learning Part I. Characterization of the model and its parameters. Comput. J. **6**, 232–236 (1963)

Minsky, M., Papert, S.A.: Perceptrons: An Introduction to Computational Geometry. MIT Press, England (1969)

Miranda, E., Aryuni, M., Irwansyah, E.: A survey of medical image classification techniques. In: International Conference on Information Management and Technology, pp. 56–61 (2016)

Mitchell, T.M.: Machine Learning. McGraw-Hill Higher Education, New York (1997)

Morgan, J.N., Sonquist, J.A.: Problems in the analysis of survey data, and a proposal. J. Am. Stat. Assoc. **58**, 415–434 (1963)

Müller, K.R., Smola, A.J., Rätsch, G., Schölkopf, B., Kohlmorgen, J., Vapnik, V.: Predicting time series with support vector machines. In: International Conference on Artificial Neural Networks, pp. 999–1004. Springer, Heidelberg (1997)

Nilsson, N.J.: Learning Machines. McGraw-Hill, New York (1965)

Pan, Z., Yu, W., Yi, X., Khan, A., Yuan, F., Zheng, Y.: Recent progress on generative adversarial networks (GANs): a survey. IEEE Access **7**, 36322–36333 (2019)

Pavlidis, P., Weston, J., Cai, J., Grundy, W N.: Gene functional classification from heterogeneous data. In: Proceedings of the Fifth Annual International Conference on Computational Biology, pp. 249–255 (2001)

Poggio, T.: On optimal nonlinear associative recall. Biol. Cybern. **19**, 201–209 (1975)

Poggio, T., Girosi, F.: Networks for approximation and learning. Proc. IEEE **78**, 1481–1497 (1990)

Polson, N.G., Scott, S.L.: Data augmentation for support vector machines. Bayesian Anal. **6**, 1–23 (2011)

Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. Psychol. Rev. **65**, 386–408 (1958)

Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**, 533–536 (1986)

Samuel, A.L.: Some studies in machine learning using the game of checkers. IBM J. Res. Dev. **3**, 210–229 (1959)

Samuel, A.L.: Some studies in machine learning using the game of checkers. IBM J. Res. Dev. **44**, 206–226 (2000)

Schapire, R.E.: The strength of weak learnability. Mach. Learn. **5**, 197–227 (1990)

Sebestyen, G.S.: Decision-Making Processes in Pattern Recognition. Macmillan, New York (1962)

Shawe-Taylor, J., Cristianini, N.: Margin distribution and soft margin. In: Smola, A.J., Bartlett, P., Scholkopf, B., Schuurmans, D., (eds.), Advances in Large Margin Classifiers, pp. 349–358. MIT Press, England (2000)

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y.: Mastering the game of go without human knowledge. Nature **550**, 354–359 (2017)

Smith, F.W.: Pattern classifier design by linear programming. IEEE Trans. Comput. **100**, 367–372 (1968)

Solomonoff, R.J.: A formal theory of inductive inference. Part II. Inf. Control. **7**, 224–254 (1964)

Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, USA (2018)

Tuia, D., Volpi, M., Copa, L., Kanevski, M., Munoz-Mari, J.: A survey of active learning algorithms for supervised remote sensing image classification. IEEE J. Sel. Topics Signal Process. **5**, 606–617 (2011)

Vapnik, V.: Pattern recognition using generalized portrait method. Autom. Remote. Control. **24**, 774–780 (1963)

Vapnik, V.N.: Estimation of Dependencies Based on Empirical Data. Springer, New York (1982)

Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995)

Vapnik, V.N., Chervonenkis, A.Y.: On a class of perceptrons. Autom. Remote. **25**, 103–109 (1964)

Vapnik, V., Chervonenkis, A.: Theory of pattern recognition: statistical problems of learning (Russian). Nauka, Moscow (1974)

Vapnik, V., Chervonenkis, A.: Theory of Pattern Recognition (German). Akademie, Berlin (1979)

Wahba, G.: Spline Models for Observational Data. SIAM, PA (1990)

Warwick, K.: A Brief History of Deep Blue. IBM's Chess Computer, Mental Floss (2017)

Watkins, C.J.C.H.: Learning from delayed rewards. Ph.D. thesis, University of Cambridge, England (1989)

Wenzel, F., Galy-Fajou, T., Deutsch, M., Kloft, M.: Bayesian nonlinear support vector machines for big data. In: European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 307–322. Springer, Cham (2017)

Widrow, B., Hoff, M.E.: Adaptive Switching Circuits (No. TR-1553-1). Stanford University California Stanford Electronics Labs (1960)

Wolpert, D.H.: The lack of a priori distinctions between learning algorithms. Neural Comput. **8**, 1341–1390 (1996)

Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Trans. Neural Netw. Learn. Syst. **16**, 645–678 (2005)