support vector machine (SVM)

Authors

Alexander Jung*, Department of Computer Science, Aalto University, Espoo, Finland
*Corresponding author: alex.jung@aalto.fi

Synonyms

maximum-margin classifier

Abstract

The support vector machine (SVM) is a binary classification method that learns a linear classifier by regularized empirical risk minimization (RERM), combining the hinge loss with a squared-norm penalty term. For a linearly separable training set and sufficiently weak regularization, the solution is the maximum-margin separating hyperplane. This hyperplane is fully determined by the feature vectors closest to it, which are called support vectors. The kernel SVM is obtained by combining the basic SVM with a feature transformation derived from a kernel function.

Definition

The support vector machine (SVM) is a binary classification method for data points with feature space $\mathcal{X} = \mathbb{R}^d$. In its simplest form, the SVM learns a linear classifier with decision boundary

$$\{\mathbf{x} \in \mathbb{R}^d : \mathbf{w}^\top \mathbf{x} + b = 0\}.$$

The linear classifier is parameterized by a nonzero vector $\mathbf{w} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$ and an offset $b \in \mathbb{R}$. The vector $\mathbf{w}$ is the normal vector to the decision boundary and the offset b shifts the decision boundary away from the origin.

For a data point with feature vector $\mathbf{x}$ and label $y \in \{-1, +1\}$, the SVM delivers the prediction $\widehat{y} = \text{sign}(\mathbf{w}^\top \mathbf{x} + b)$. SVMs have been applied to text classification, where each document is first represented by a numeric feature vector of word frequencies and then categorized by topic. Another application is handwritten digit recognition, where digit images are classified from pixel-level features (Cristianini and Shawe-Taylor, 2000).

The SVM can be formulated as an instance of regularized empirical risk minimization (RERM) for learning a linear classifier from a training set that consists of m data points. The model parameters $(\widehat{\mathbf{w}}, \widehat{b})$ of the linear classifier are obtained as

$$(\widehat{\mathbf{w}}, \widehat{b}) = \underbrace{\operatorname*{arg\,min}_{\mathbf{w} \in \mathbb{R}^d,\, b \in \mathbb{R}} \frac{1}{m} \sum_{r=1}^{m} \max\{0,\, 1 - y^{(r)}(\mathbf{w}^\top \mathbf{x}^{(r)} + b)\} + \alpha \|\mathbf{w}\|_2^2}_{:= f(\mathbf{w}, b)}. \quad (1)$$

The SVM objective function consists of the average hinge loss on the training set plus a penalty term $\alpha\|\mathbf{w}\|_2^2$ with regularization parameter $\alpha > 0$ (Hastie et al., 2009, Sect. 12.3.2, p. 426; Shalev-Shwartz and Ben-David, 2014, Sect. 15.2, Eq. (15.5)). The penalty term in (1) is the scaled squared Euclidean norm $\alpha\|\mathbf{w}\|_2^2$, the same penalty term as in ridge regression. It does not involve the model parameter b, which is therefore left unpenalized.

In linear regression, the penalty term can be interpreted as an estimate of how much higher the loss is on data points outside the training set than on it (Hastie et al., 2009, Ch. 7; see linear regression). This interpretation rests on the quadratic structure of the squared error loss and does not carry over to the hinge loss.

For the SVM, the penalty term instead controls the generalization gap: the hinge loss is Lipschitz continuous in $\mathbf{w}$, with a constant given by the Euclidean norm of the feature vector. Adding the penalty term $\alpha\|\mathbf{w}\|_2^2$ to the average hinge loss has two effects: First, it makes the overall objective function in (1) strongly convex in $\mathbf{w}$, so that its solution $\widehat{\mathbf{w}}$ is unique. Second, it controls how much that solution varies if the data points in the training set change.

If the feature vectors in the training set have bounded Euclidean norm, the expectation of the difference between the risk and the empirical hinge loss of the learned linear classifier decreases in proportion to $1/(\alpha m)$ (Shalev-Shwartz and Ben-David, 2014, Cor. 13.6, Cor. 15.7). The penalty term also has a geometric role: a small norm $\|\widehat{\mathbf{w}}\|_2$ corresponds to a large margin $1/\|\widehat{\mathbf{w}}\|_2$ of the learned linear classifier, as discussed below.

Like logistic regression and linear regression, the SVM learns the model parameters $(\mathbf{w}, b)$ of a linear model. In contrast to logistic regression and linear regression, the SVM objective function $f(\mathbf{w}, b)$ is non-smooth because of the hinge loss. However, since the SVM objective function is convex, the optimization problem (1) can be solved by convex optimization methods. One such method is subgradient descent, which is obtained from gradient descent (GD) by replacing the gradient of the objective function with a subgradient (Shor, 1985; Bertsekas, 2016; Boyd and Vandenberghe, 2004).

Starting from an initial choice $\mathbf{w}^{(0)}, b^{(0)}$ of the model parameters, subgradient descent repeatedly applies the update

$$\big(\mathbf{w}^{(t+1)}, b^{(t+1)}\big) = \big(\mathbf{w}^{(t)}, b^{(t)}\big) - \eta^{(t)}\mathbf{g}^{(t)}, \text{ for } t = 0, 1, \ldots,$$

with a step size $\eta^{(t)} > 0$ and a subgradient $\mathbf{g}^{(t)} \in \partial f\big(\mathbf{w}^{(t)}, b^{(t)}\big)$ of the SVM objective function (1). Inserting the SVM objective function (1) into the generic update yields the explicit update

$$\mathbf{w}^{(t+1)} = \big(1 - 2\eta^{(t)}\alpha\big)\mathbf{w}^{(t)} + \frac{\eta^{(t)}}{m}\sum_{r=1}^{m}\beta_r^{(t)} y^{(r)}\mathbf{x}^{(r)},$$

$$b^{(t+1)} = b^{(t)} + \frac{\eta^{(t)}}{m}\sum_{r=1}^{m}\beta_r^{(t)} y^{(r)},$$

with expansion coefficients $\beta_r^{(t)} \in [0, 1]$, for $r = 1, \ldots, m$. The coefficient $\beta_r^{(t)}$ is

nonzero only for data points that satisfy $y^{(r)}\left(\left(\mathbf{w}^{(t)}\right)^\top \mathbf{x}^{(r)} + b^{(t)}\right) \leq 1$, i.e., data points on which the hinge loss is nonzero or non-differentiable at the current model parameters. Each iteration scales $\mathbf{w}^{(t)}$ by the factor $1 - 2\eta^{(t)}\alpha$ (the effect of the penalty term) and adds corrections $y^{(r)}\mathbf{x}^{(r)}$ only from these data points. In particular, for the initialization $\mathbf{w}^{(0)} = \mathbf{0}$, every iterate $\mathbf{w}^{(t)}$ is a weighted sum $\sum_{r=1}^{m} \beta_r y^{(r)} \mathbf{x}^{(r)}$ of the feature vectors in the training set. The solution $\widehat{\mathbf{w}}$ of (1) admits the same expansion, with coefficients that are nonzero only for the data points satisfying the same condition $y^{(r)}\left(\widehat{\mathbf{w}}^\top \mathbf{x}^{(r)} + \widehat{b}\right) \leq 1$ (see the discussion of support vectors below). For a suitably diminishing step size, e.g., $\eta^{(t)} = 1/(t+1)$, the iterates $\left(\mathbf{w}^{(t)}, b^{(t)}\right)$ converge to a solution of (1) (Shor, 1985).

The solution of (1) has a clear geometric meaning when the training set is linearly separable, i.e., there is some choice for $(\mathbf{w}, b)$ such that $\operatorname{sign}(\mathbf{w}^\top \mathbf{x}^{(r)} + b) y^{(r)} = 1$ for $r = 1, \ldots, m$. For a sufficiently small α, the solution of (1) determines the separating hyperplane that is farthest from the feature vectors in the training set (see Fig. 1). The distance between this hyperplane and the nearest feature vectors, which are precisely the support vectors discussed below, is referred to as the margin and is given by $1/\|\widehat{\mathbf{w}}\|_2$ (Boser et al., 1992; Cortes and Vapnik, 1995; Cristianini and Shawe-Taylor, 2000).

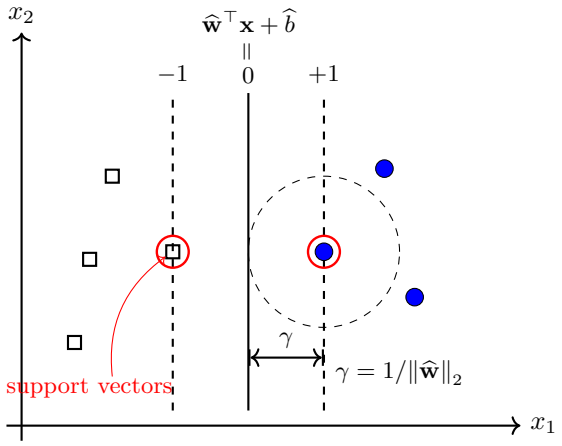

Figure 1: Maximum-margin separation for a linearly separable training set. The decision boundary $\widehat{\mathbf{w}}^\top \mathbf{x} + \widehat{b} = 0$ (solid line) lies halfway between two parallel hyperplanes $\widehat{\mathbf{w}}^\top \mathbf{x} + \widehat{b} = \pm 1$ (dashed lines). The support vectors (open square and filled circle inside red rings) are the feature vectors that lie on these hyperplanes. The margin $\gamma = 1/\|\widehat{\mathbf{w}}\|_2$ is the distance from the decision boundary to the support vectors. Data points with label $y = +1$ are drawn as filled circles, those with $y = -1$ as open squares. The dashed circle of radius γ shows that any support vector can be displaced by a perturbation with Euclidean norm less than γ without crossing the decision boundary

The margin of the learned hyperplane measures robustness against perturbations of the features of a data point. Perturbing a feature vector $\mathbf{x}$ by a vector $\boldsymbol{\delta}$ changes the score $\widehat{\mathbf{w}}^\top \mathbf{x} + \widehat{b}$ by $\widehat{\mathbf{w}}^\top \boldsymbol{\delta}$, whose magnitude is at most $\|\widehat{\mathbf{w}}\|_2 \|\boldsymbol{\delta}\|_2$ by the Cauchy-Schwarz inequality. A data point therefore stays correctly classified under every feature perturbation with $\|\boldsymbol{\delta}\|_2 < \gamma$, as illustrated by the dashed circle around a support vector in Fig. 1. Data points whose feature vectors lie farther from the decision boundary tolerate even larger perturbations: the tolerated radius equals the distance of the feature vector from the decision boundary, which is at least γ and equals γ precisely for the support vectors. Maximizing the margin maximizes this worst-case tolerated radius, so the maximum-margin classifier is the separating classifier that is robust against the largest feature perturbations (Xu et al., 2009).

While the above discussion only applies when the SVM training set is linearly separable, every solution of the SVM problem (1) has a structure that holds in general. In particular,

$$\widehat{\mathbf{w}} = \sum_{r=1}^{m} \beta_r y^{(r)} \mathbf{x}^{(r)}$$

with expansion coefficients $0 \leq \beta_r \leq 1/(2\alpha m)$ that are nonzero only for the feature vectors with

$$y^{(r)}(\widehat{\mathbf{w}}^\top \mathbf{x}^{(r)} + \widehat{b}) \leq 1.$$

These feature vectors are referred to as support vectors since they entirely determine the SVM solution $(\widehat{\mathbf{w}}, \widehat{b})$ (Boser et al., 1992; Cortes and Vapnik, 1995). Applying a perturbation to any feature vector which is not a support vector leaves $\widehat{\mathbf{w}}$, $\widehat{b}$ unchanged, provided the perturbation is small enough to preserve the strict inequality $y(\widehat{\mathbf{w}}^\top \mathbf{x} + \widehat{b}) > 1$ (see Fig. 2).

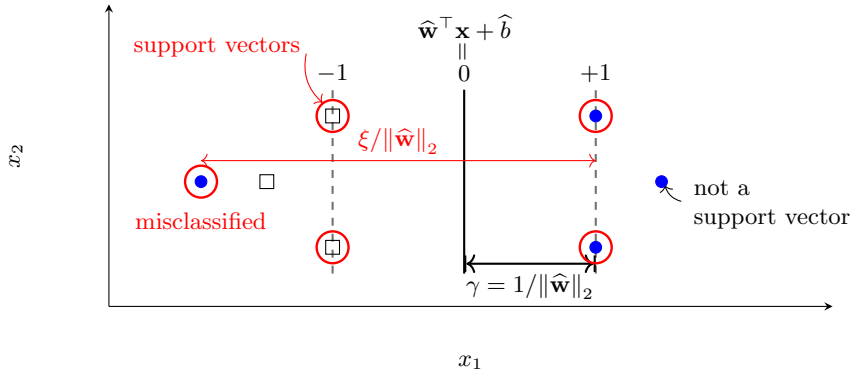

Figure 2: Non-separable training set of $m = 7$ data points in feature space $\mathbb{R}^2$. The solution of the SVM problem (1) places the decision boundary $\widehat{\mathbf{w}}^\top \mathbf{x} + \widehat{b} = 0$ between the two hyperplanes $\widehat{\mathbf{w}}^\top \mathbf{x} + \widehat{b} = \pm 1$ at distance $\gamma = 1/\|\widehat{\mathbf{w}}\|_2$. These two hyperplanes contain four of the support vectors (red rings), given by the data points with margin equal to γ. The outlier, labeled $+1$, is also a support vector; it is misclassified beyond the opposite margin: $y(\widehat{\mathbf{w}}^\top \mathbf{x} + \widehat{b}) = -2 < -1$, with hinge loss value ξ. The two data points (open square and filled circle not inside red rings) farther from the decision boundary than the margin γ are not support vectors. Data generated by `pythondemos/svm.py`

An overly large regularization parameter α makes the SVM underfit: the above expansion of $\widehat{\mathbf{w}}$ implies $\|\widehat{\mathbf{w}}\|_2 \leq \rho/(2\alpha)$, with ρ the largest Euclidean norm of a feature vector in the training set, so $\widehat{\mathbf{w}}$ shrinks toward $\mathbf{0}$.

Consider a training set with unequal class sizes. For sufficiently large α, the predictions approach the constant majority-class prediction $\text{sign}(\widehat{b})$ and every data point of the minority class is a misclassified support vector. The training set of Fig. 2 also has unequal class sizes, four data points labeled $y = +1$ and three labeled $y = -1$, but with the value $\alpha = 1/14$ used there, the SVM misclassifies one data point which can be considered an outlier. Increasing the value of α to 50 for that training set shrinks $\|\widehat{\mathbf{w}}\|_2$, makes every prediction on the training set equal to $+1$, and misclassifies all three data points of the minority class.

The basic SVM (1) learns a linear classifier on the feature space $\mathcal{X} = \mathbb{R}^d$. It can be generalized to a classification method for data points whose feature vectors lie in an arbitrary feature space $\mathcal{X}$ by first applying a feature transformation $\phi : \mathcal{X} \to \mathcal{X}'$ with $\mathcal{X}' = \mathbb{R}^d$ and then learning a linear classifier on the transformed feature vectors. By choosing ϕ (and d) appropriately, any given training set can be made linearly separable in $\mathcal{X}'$ (Cortes and Vapnik, 1995; Cristianini and Shawe-Taylor, 2000). A principled construction of a feature transformation ϕ is via a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. In this construction, the transformed feature space is, in general, not $\mathbb{R}^d$ but a Hilbert space $\mathcal{H}$, which can be infinite-dimensional (see kernel method). The resulting method is then referred to as a kernel SVM (Schölkopf and Smola, 2002).

Cross References

binary classification, linear model, classifier, hinge loss, margin, hyperplane, decision boundary, robustness, ridge regression, Lasso, kernel, kernel method, subgradient descent

References

1. Bertsekas (2016). Nonlinear Programming. 3rd ed., Athena Scientific, Belmont, MA, USA.

2. Boser et al. (1992). A Training Algorithm for Optimal Margin Classifiers. In: Proc. 5th Annual Workshop on Computational Learning Theory (COLT), pp. 144–152. ACM.

3. Boyd and Vandenberghe (2004). Convex Optimization. Cambridge Univ. Press, Cambridge, U.K.

4. Cristianini and Shawe-Taylor (2000). An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge Univ. Press, New York, NY, USA.

5. Schölkopf and Smola (2002). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA, USA.

6. Shalev-Shwartz and Ben-David (2014). Understanding Machine Learning: From Theory to Algorithms. Cambridge Univ. Press, New York, NY, USA.

7. Shor (1985). Minimization Methods for Non-differentiable Functions. Springer, Berlin, Germany.

8. Xu et al. (2009). Robustness and Regularization of Support Vector Machines. J. Mach. Learn. Res., 10(51), pp. 1485–1510.

9. Cortes and Vapnik (1995). Support-vector networks. Machine learning, 20(3), pp. 273–297.

10. Hastie et al. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed., Springer Science+Business Media, New York, NY, USA.