Support vector machines

Victor Kitov

v.v.kitov@yandex.ru

Yandex
School of Data Analysis.

Table of Contents

Support vector machines

(a) (b)

Support vector machines

(a) (b)

Main idea

Select hyperplane maximizing the spread between classes.

Support vector machines

Objects x_i for $i = 1, 2, ...n$ lie at distance $b/|w|$ from discriminant hyperplane if

$$\begin{cases} x_i^T w + w_0 \ge b, & y_i = +1 \\ x_i^T w + w_0 \le -b & y_i = -1 \end{cases} \quad i = 1, 2, ...N.$$

This can be rewritten as

$$y_i(x_i^T w + w_0) \ge b, \quad i = 1, 2, ...N.$$

The margin is equal to $2b/\|w\|$. Since w, w_0 and b are defined up to multiplication constant, we can set $b = 1$.

Problem statement

Problem statement:

$$\begin{cases} \frac{1}{2} w^T w \to \min\limits_{w, w_0} \\ y_i(x_i^T w + w_0) \geq 1, \quad i = 1, 2, \ldots N. \end{cases}$$

Support vectors

non-informative observations: $y_i(x_i^T w + w_0) > 1$

- do not affect the solution

support vectors: $y_i(x_i^T w + w_0) = 1$

- lie at distance $1/\|w\|$ to separating hyperplane
- affect the the solution.

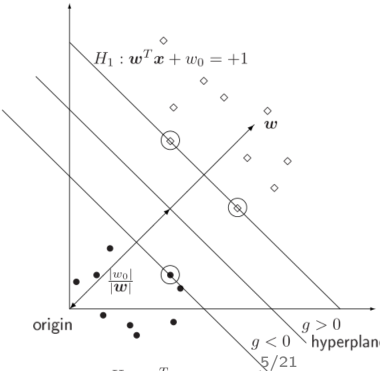
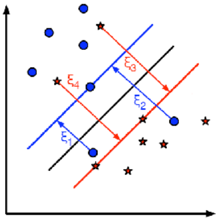
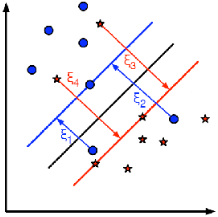


Table of Contents

Linearly non-separable case

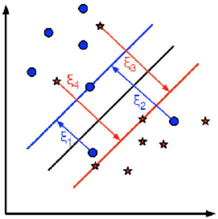


Linearly non-separable case



$$\begin{cases} \frac{1}{2}w^T w \to \min_{w,w_0} \\ y_i(x_i^T w + w_0) \geq 1, \quad i = 1, 2, \ldots N. \end{cases}$$

Linearly non-separable case



$$\begin{cases} \frac{1}{2}w^T w \to \min_{w, w_0} \\ y_i(x_i^T w + w_0) \ge 1, \quad i = 1, 2, ...N. \end{cases}$$

Problem

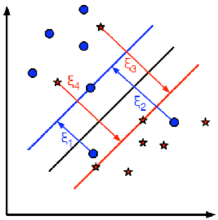
Constraints become incompatible and give empty set!

Linearly non-separable case

No separating hyperplane exists. Errors are permitted by including slack variables ξ_i:

$$\begin{cases} \frac{1}{2}w^T w + C\sum_{i=1}^{N}\xi_i \to \min_{w,\xi} \\ y_i(w^T x_i + w_0) \geq 1 - \xi_i,\ i = 1, 2, ...N \\ \xi_i \geq 0,\ i = 1, 2, ...N \end{cases}$$

- Parameter C is the cost for misclassification and controls the bias-variance trade-off.
- It is chosen on validation set.
- Other penalties are possible, e.g. $C\sum_i \xi_i^2$.



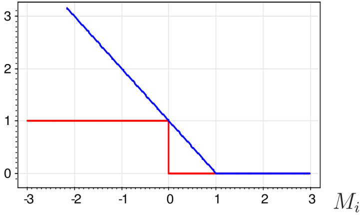
Classification of training objects

- **Non-informative objects**:
    - $y_i(w^T x_i + w_0) > 1$
- **Support vectors** SV**:**
    - $y_i(w^T x_i + w_0) \leq 1$
    - **boundary support vectors** $\widetilde{SV}$:
        - $y_i(w^T x_i + w_0) = 1$
    - **violating support vectors**:
        - $y_i(w^T x_i + w_0) > 0$: violating support vector is correctly classified.
        - $y_i(w^T x_i + w_0) < 0$: violating support vector is misclassified.

SVM with unconstrained optimization

Optimization problem:

$$\begin{cases} \frac{1}{2}w^T w + C\sum_{i=1}^{N}\xi_i \to \min_{w,w_0,\xi} \\ y_i(w^T x_i + w_0) = M_i(w,w_0) \geq 1-\xi_i, \\ \xi_i \geq 0,\ i=1,2,...N \end{cases}$$



can be rewritten as

$$\frac{1}{2C}\|w\|_2^2 + \sum_{i=1}^{N}[1-M_i(w,w_0)]_+ \to \min_{w,w_0,\xi}$$

Thus SVM is linear discriminant function with cost approximated with $\mathcal{L}(M) = [1-M]_+$ and L_2 regularization.

Sparsity of solution

- SVM solution depends only on support vectors
- This is also clear from loss function, satisfying $\mathcal{L}(M) = 0$ for $M \geq 1$.
  - objects with margin≥ 1 don't affect solution!
- Sparsity causes SVM to be less robust to outliers
  - because outliers are always support vectors

Multiclass SVM

C discriminant functions are built simultaneously:

$$g_c(x) = (\mathbf{w}^c)^T x + w_0^c, \qquad c = \overline{1, C}.$$

Linearly separable case:

$$\begin{cases} \sum_{c=1}^{C} (\mathbf{w}^c)^T \mathbf{w}^c \to \min_{\mathbf{w}} \\ (\mathbf{w}^{y_n})^T x_n + w_0^{y_n} - (\mathbf{w}^c)^T x - w_0^c \geq 1 \quad \forall c \neq y_n, \\ n = \overline{1, N}. \end{cases}$$

Linearly non-separable case:

$$\begin{cases} \sum_{c=1}^{C} (\mathbf{w}^c)^T \mathbf{w}^c + C \sum_{n=1}^{N} \xi_n \to \min_w \\ (\mathbf{w}^{y_n})^T x + w_0^{y_n} - (\mathbf{w}^c)^T x - w_0^c \geq 1 - \xi_n \quad \forall c \neq y_n, \\ \xi_n \geq 0, \quad n = \overline{1, N}. \end{cases}$$

Is slower, but shows similar accuracy to one-vs-all,one-vs-one SVM.

Table of Contents

Dual problem

Solving Karush-Kuhn-Takker conditions, get **dual optimization problem**:

$$\begin{cases} L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j \to \max_\alpha \\ \sum_{n=1}^{N} \alpha_n y_n = 0 \\ 0 \le \alpha_n \le C, \quad n = \overline{1, N} \end{cases} \tag{1}$$

It is standard quadratic programming task.

Comments on support vectors

- **non-informative vectors**: $y_i(w^T x_i + w_0) > 1$ have $\alpha_i = 0$
- **non-boundary support vectors** $SV \setminus \tilde{SV}$:
  $y_i(w^T x_i + w_0) < 1$ have $\alpha_i = C$.
- **boundary support vectors** $\widetilde{\mathcal{SV}}$: $y_i(w^T x_i + w_0) = 1$
  Typically $\alpha_i \in (0, C)$, though $\alpha_i = 0, C$ are possible as special cases.

Solution

1. Solve (1) to find optimal dual variables α_i^*
2. Find optimal w ($\alpha_i^* \neq 0$ only for support vectors):

$$w = \sum_{i \in \mathcal{SV}} \alpha_i^* y_i x_i$$

3. w_0 can be found from any edge equality for boundary support vector:

$$y_i(x_i^T w + w_0) = 1, \forall i \in \widetilde{\mathcal{SV}} \qquad (2)$$

Solution for w_0

By multiplyting (2) by y_i obtain

$$x_i^T w + w_0 = y_i \quad \forall i \in \widetilde{\mathcal{SV}} \tag{3}$$

Get more numerically stable from summing 3 over all $i \in \widetilde{\mathcal{SV}}$:

$$n_{\tilde{SV}} w_0 = \sum_{j \in \tilde{SV}} \left(y_j - x_j^T w \right) = \sum_{j \in \tilde{SV}} y_j - \sum_{j \in \tilde{SV}} x_j^T w, \quad n_{\tilde{SV}} = \left| \tilde{SV} \right|$$

$$w_0 = \frac{1}{n_{\tilde{SV}}} \left(\sum_{j \in \tilde{SV}} y_j - \sum_{j \in \tilde{SV}} \overbrace{\sum_{i \in \mathcal{SV}} \alpha_i^* y_i x_i^T}^{w^T} x_j \right)$$

If there exist no boundary support vectors (only violating SV), then find w_0 by grid search.

Making predictions

1. Solve dual task to find α_i^*, $i = 1, 2, \dots N$

$$\begin{cases} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \to \max_\alpha \\ \sum_{i=1}^N \alpha_i y_i = 0 \\ 0 \le \alpha_i \le C \quad \text{(using (??) and that } \alpha_i \ge 0,\ r_i \ge 0) \end{cases}$$

2. Find optimal w_0:

$$w_0 = \frac{1}{n_{\tilde{SV}}} \left(\sum_{j \in \tilde{SV}} y_j - \sum_{j \in \tilde{SV}} \sum_{i \in \mathcal{SV}} \alpha_i^* y_i \langle x_i, x_j \rangle \right)$$

3. Make prediction for new x:

$$\widehat{y} = \text{sign}[w^T x + w_0] = \text{sign}[\sum_{i \in \mathcal{SV}} \alpha_i^* y_i \langle x_i, x \rangle + w_0]$$

Making predictions

1. Solve dual task to find α_i^*, $i = 1, 2, ... N$

$$\begin{cases} L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2}\sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \to \max_\alpha \\ \sum_{i=1}^N \alpha_i y_i = 0 \\ 0 \le \alpha_i \le C \quad \text{(using (??) and that } \alpha_i \ge 0,\ r_i \ge 0) \end{cases}$$

2. Find optimal w_0:

$$w_0 = \frac{1}{n_{\tilde{SV}}} \left(\sum_{j \in \tilde{SV}} y_j - \sum_{j \in \tilde{SV}} \sum_{i \in \mathcal{SV}} \alpha_i^* y_i \langle x_i, x_j \rangle \right)$$

3. Make prediction for new x:

$$\hat{y} = \text{sign}[w^T x + w_0] = \text{sign}[\sum_{i \in \mathcal{SV}} \alpha_i^* y_i \langle x_i, x \rangle + w_0]$$

- On all steps we don't need exact feature representations, only scalar products $\langle x, x' \rangle$!

Kernel trick generalization

1. Solve dual task to find α_i^*, $i = 1, 2, \dots N$

$$\begin{cases} L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) \to \max_\alpha \\ \sum_{i=1}^N \alpha_i y_i = 0 \\ 0 \le \alpha_i \le C \end{cases}$$

2. Find optimal w_0:

$$w_0 = \frac{1}{n_{\tilde{SV}}} \left(\sum_{j \in \tilde{SV}} y_j - \sum_{j \in \tilde{SV}} \sum_{i \in \mathcal{SV}} \alpha_i^* y_i K(x_i, x_j) \right)$$

3. Make prediction for new x:

$$\widehat{y} = \text{sign}[w^T x + w_0] = \text{sign}[\sum_{i \in \mathcal{SV}} \alpha_i^* y_i K(x_i, x) + w_0]$$

- We replaced $\langle x, x' \rangle \to K(x, x')$ for $K(x, x') = \langle \phi(x), \phi(x') \rangle$ for some feature transformation $\phi(\cdot)$.

Summary

- SVM - linear classifier with L_2 regularization and hinge loss.
- Geometrically SVM maximizes border between classes.
- Solution depends only on support vectors, having margin≤ 1.
- Solution depends on x only through $\langle x_i, x_j \rangle$
    - may generalize $\langle x_i, x_j \rangle$ to $K(x_i, x_j)$.