# Support vector machines

Victor Kitov

# Table of contents

# Kuhn-Takker conditions

Consider the optimization task:

$$\begin{cases} f(x) \to \min_x \\ g_i(x) \le 0 \qquad i = 1, 2, ...m \end{cases} \tag{1}$$

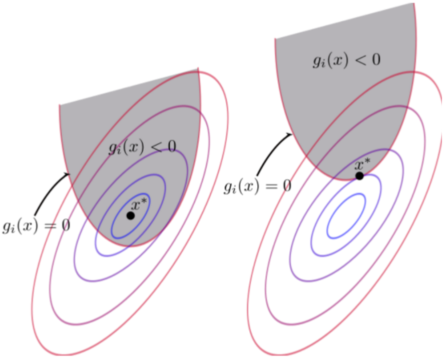
**Theorem (necessary conditions for optimality):**
Let

- $x^*$ - be the solution to (1),
- $f(x^*)$ and $g_i(x^*)$, $i = 1, 2, ...m$ - continuously differentiable at $x^*$.
- one of the conditions of regularity is satisfied

Then coefficients $\lambda_1, \lambda_2, ...\lambda_m$ exist, such that $x^*$ satisfies the conditions:

$$\begin{cases} \nabla f(x^*) + \sum_{i=1}^m \lambda_i \nabla g_i(x^*) = 0 & \text{stationarity} \\ g_i(x^*) \le 0 & \text{feasibility} \\ \lambda_i \ge 0 & \text{non-negativity} \\ \lambda_i g_i(x^*) = 0 & \text{complementary slackness} \end{cases} \tag{2}$$

# Illustration of constrained optimization



# Kuhn-Takker conditions

**Possible regularity conditions:**

- $\{\nabla g_j(x^*), j \in J\}$ - linearly independent, where $J$ are indexes of active constraints $J = \{j : g_j(x^*) = 0\}$.
- Slater condition: $\exists x : \ g_i(x) < 0 \, \forall i$ (applicable only when $f(x)$ and $g_i(x)$, $i = 1, 2, ...m$ are convex)

**Sufficient conditions of optimality:**
If $f(x)$ and $g_i(x)$, $i = 1, 2, ...m$ are convex, Kuhn-Takker conditions (2) and Slater conditions become sufficient for $x^*$ to be the solution of (1).

# Convex optimization

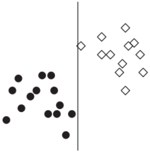
Why convexity of $f(x)$ and $g_i(x)$, $i = 1, 2, ...m$ is convenient:

- All local minimums become global minimums
- The set of minimums is convex
- If $f(x)$ is strictly convex and minimum exists, then it is unique.

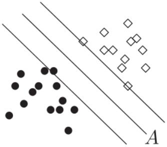
# Table of contents

2. Support vector machines
   - Linearly separable case
   - Linearly non-separable case

# 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

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

This can be rewritten as

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

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:

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

# Problem statement

Problem statement:

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

Lagrangian:

$$L = \frac{1}{2}w^T w - \sum_{i=1}^{N} \alpha_i (y_i (w^T x + w_0) - 1)$$

By Karush-Kuhn-Takker the solution satisfies:

$$\begin{cases} \frac{\partial L}{\partial w} = \mathbf{0}, \ \frac{\partial L}{\partial w_0} = 0 \\ y_i(x_i^T w + w_0) - 1 \geq 0, \\ \alpha_i (y_i (x_i^T w + w_0) - 1) = 0, \\ \alpha_i \geq 0, \quad i = 1, 2, ...N \end{cases}$$

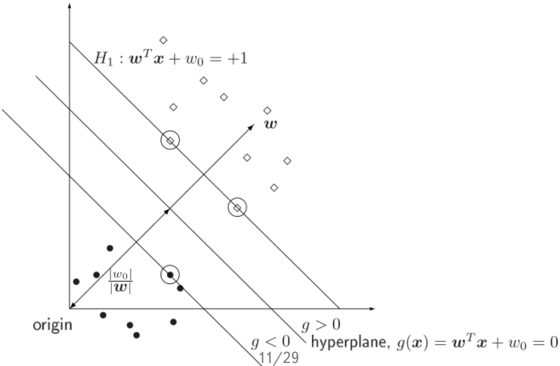
# Support vectors

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

- do not affect the solution

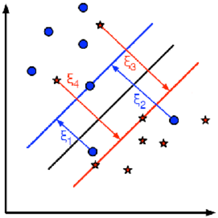
**support vectors:** $y_i(x_i^T w + w_0) = 1$

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

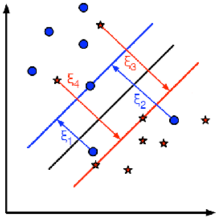


2. Support vector machines
    - Linearly separable case
    - Linearly non-separable case

# Linearly non-separable case

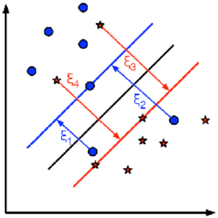


# Linearly non-separable case




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

# Linearly non-separable case




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

**Problem**

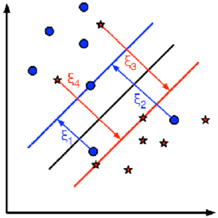
Constraints become incompatible and give empty set!

# Linearly non-separable case

No separating hyperplane exists. Errors are permitted by including slack variables $\xi_i$:

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

- 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. $C\sum_i \xi_i^2$.



# Linearly non-separable case

Lagrangian:

$$L = \frac{1}{2}w^T w + C\sum_i \xi_i - \sum_{i=1}^{N} \alpha_i(y_i(w^T x_i + w_0) - 1 + \xi_i) - \sum_{i=1}^{N} r_i \xi_i$$

By Karush-Kuhn-Takker conditions, the solution satisfies constraints:

$$\begin{cases} \frac{\partial L_P}{\partial w} = \mathbf{0},\ \frac{\partial L_P}{\partial w_0} = 0,\ \frac{\partial L_P}{\partial \xi_i} = 0 \\ \xi_i \geq 0,\ \alpha_i \geq 0,\ r_i \geq 0 \\ y_i(x_i^T w + w_0) \geq 1 - \xi_i, \\ \alpha_i(y_i(w^T x_i + w_0) - 1 + \xi_i) = 0 \\ r_i \xi_i = 0, \quad i = 1, 2, \ldots N \end{cases}$$

# Classification of training objects

- **Non-informative objects:**
    - $y_i(w^T x_i + w_0) > 1$
- **Support vectors $SV$:**
    - $y_i(w^T x_i + w_0) \le 1$
    - **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.

# Solving Karush-Kuhn-Takker conditions

$$\frac{\partial L}{\partial w}=\mathbf{0}:\ w=\sum_{i=1}^{N}\alpha_i y_i x_i \tag{3}$$

$$\frac{\partial L}{\partial w_0}=0:\ \sum_{i=1}^{N}\alpha_i y_i=0$$

$$\frac{\partial L}{\partial \xi_i}=0:\ C-\alpha_i-r_i=0 \tag{4}$$

Substituting these constraints into $L$, we obtain the *dual problem*[1]:

$$\begin{cases} L_D=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j x_i^T x_j \to \max_\alpha \\ \sum_{i=1}^{N}\alpha_i y_i=0 \\ 0\le\alpha_i\le C \quad \text{(using (4) and that } \alpha_i\ge 0,\ r_i\ge 0) \end{cases} \tag{5}$$

[1] Dual Lagrangian is maximized because original Lagrangian has saddlepoint in optimum, min for $w, w_0, \xi_i$ and max for $\alpha_i, r_i$.

# Comments on support vectors

- non support vectors: $y_i(w^T x_i + w_0) > 1$ <=> $\xi_i = 0$, $y_i(w^T x_i + w_0) - 1 + \xi_i > 0$ => $\alpha_i = 0$
  - support vectors $SV$ will have $\alpha_i > 0$.
- non-boundary support vectors $SV \setminus \tilde{SV}$: $y_i(w^T x_i + w_0) < 1$ <=> $\xi_i > 0$ => $r_i = 0$ <=> $\alpha_i = C$.
- boundary support vectors $\widetilde{\mathcal{SV}}$: $y_i(w^T x_i + w_0) = 1$ => $\xi_i = 0$
  - since $\alpha_i \in [0, C]$, $\alpha_i \in (0, C)$ for boundary support vectors.

# Solution

1. Solve (5) to find optimal dual variables $\alpha_i^*$
2. Using (3) and that $\alpha_i^* = 0$ for non support vectors, find optimal $w$

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

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

$$y_i(x_i^T w + w_0) = 1, \forall i \in \widetilde{\mathcal{SV}} \tag{6}$$

# Solution for $w_0$

By multiplyting (6) by $y_i$ obtain

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

By summing over all $i \in \widetilde{\mathcal{SV}}$ for more robust solution we obtain

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

where $n_{\tilde{SV}}$ is the number of boundary support vectors.
Finall solution for $w_0$:

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

# Making predictions

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

$$\begin{cases} \sum_{i=1}^N \alpha_i - \frac{1}{2}\sum_{i=1}^N \sum_{j=1}^N \alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle \to \max_\alpha \\ \sum_{i=1}^N \alpha_i y_i = 0 \\ 0 \le \alpha_i \le C \quad \text{(using (4) and that } \alpha_i \ge 0,\ r_i \ge 0) \end{cases}$$

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 $\alpha_i^*$, $i = 1, 2, ...N$

$$\begin{cases} L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2}\sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \to \max_\alpha \\ \sum_{i=1}^N \alpha_i y_i = 0 \\ 0 \le \alpha_i \le C \quad \text{(using (4) and that } \alpha_i \ge 0,\ r_i \ge 0) \end{cases}$$

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 $\alpha_i^*$, $i = 1, 2, ...N$

$$\begin{cases} L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2}\sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) \to \max_\alpha \\ \sum_{i=1}^N \alpha_i y_i = 0 \\ 0 \le \alpha_i \le C \quad \text{(using (4) and that } \alpha_i \ge 0,\ r_i \ge 0) \end{cases}$$

2. Find optimal $w_0$:

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

3. Make prediction for new $x$:

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

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

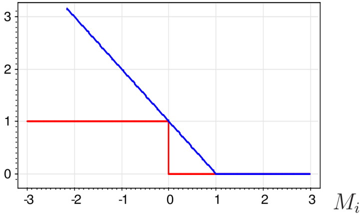
# Another view on SVM

Optimization problem:

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

can be rewritten as

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




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$\geq$ 1 don't affect solution!
- Sparsity causes SVM to be less robust to outliers
  - because outliers are always support vectors

# Multiclass classification

$C$ classes $\omega_1, \omega_2, ...\omega_C$.

- One-against-all:
  - build C binary classifiers, classifying class $\omega_i$ against other classes
  - select the class with highest margin
- One-against-one:
  - build C(C-1)/2 classifiers, classifying class $\omega_i$ against $\omega_j$.
  - select the class having maximum votes
- Multiclass variant of initial algorithm

## Table of contents

# SVM regression

Predict real-valued output with

$$\widehat{y}(x) = w^T x + w_0$$

where parameters $w, w_0$ are found from

$$\begin{cases} \left(x^T x_n + w_0\right) - y_n \le \varepsilon + \xi_n \\ y_n - \left(x^T x_n + w_0\right) \le \varepsilon + \tilde{\xi}_n \\ \xi_n, \tilde{\xi}_n \ge 0, \quad n = 1, 2, \ldots N. \\ \frac{1}{2} w^T w + C \sum_{n=1}^N \left(\xi_n + \tilde{\xi}_n\right) \to \min_{w, w_0, \xi_n, \tilde{\xi}_n} \end{cases}$$

Gives $\varepsilon$-insensitive loss!

# Multiclass SVM

$C$ discriminant functions are built simultaneously:

$$g_k(x) = (w^k)^T x + w_0^k$$

Linearly separable case:

$$\begin{cases} \sum_{k=1}^{C} (w^k)^T w^k \to \min_w \\ (w^{y(i)})^T x + w_0^{y(i)} - (w^k)^T x - w_0^k \geq 1 \quad \forall k \neq y(i), \\ i = 1, 2, ...N. \end{cases}$$

Linearly non-separable case:

$$\begin{cases} \sum_{k=1}^{C} (w^k)^T w^k + C \sum_{i=1}^{N} \xi_i \to \min_w \\ (w^{y(i)})^T x + w_0^{y(i)} - (w^k)^T x - w_0^k \geq 1 - \xi_i \quad \forall k \neq y(i), \\ \xi_i \geq 0, \quad i = 1, 2, ...N. \end{cases}$$

Is slower, but shows similar accuracy to usual SVM.