Foundations of Data Science, Fall 2020

12. Support Vector Machines I

Prof. Dan Olteanu

DaST Data•(Systems+Theory)

University of Zurich[UZH]

Nov 6, 2020

https://lms.uzh.ch/url/RepositoryEntry/16830890400

https://uzh.zoom.us/j/96690150974?pwd=cnZmMTduWUtCeWoxYW85Z3RMYnpTZz09

Support Vector Machines (SVM)

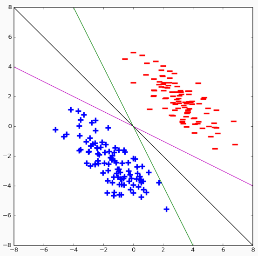
SVM is a popular discriminative model for classification

- No natural probabilistic interpretation (see one in Murphy Section 14.5.5)

SVM requires the introduction of several new concepts

- Maximum Margin Principle
- Hinge Loss optimisation
- Primal vs Dual Formulation
- Kernel Methods for non-linear classification

Binary Classification

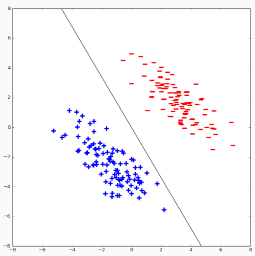


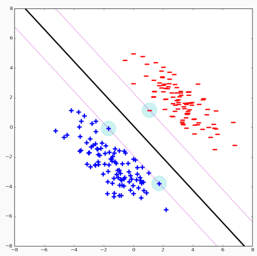
Goal: Find a linear separator

Data is linearly separable if there exists a linear separator that classifies all points correctly

Which separator should be picked?

Maximum Margin Principle





Margin for a data point = its distance to the separating hyperplane

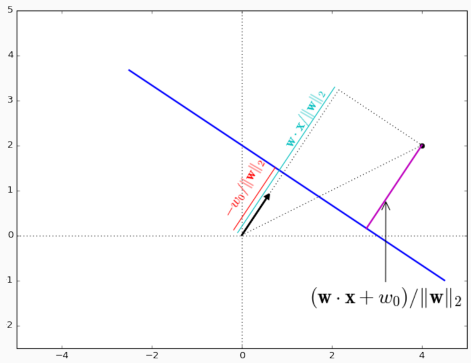
Maximum margin principle: Pick the separating boundary that maximises the smallest marging (the least distance between data and boundary)

That is: Maximise the distance of the closest point from the decision boundary

Points that are closest to the decision boundary are called support vectors

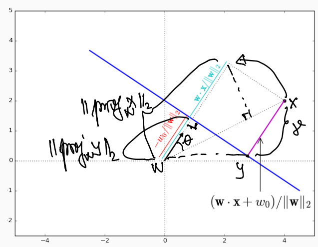
Geometry Review: Distance from Point to Hyperplane

Given a hyperplane $H \equiv \mathbf{w} \cdot \mathbf{a} + w_0 = 0$ and a point $\mathbf{x} \in \mathbb{R}^D$, how far is $\mathbf{x}$ from H?



Geometry Review: Working Out the Distance

Given a hyperplane $H \equiv \mathbf{w} \cdot \mathbf{a} + w_0 = 0$ and a point $\mathbf{x} \in \mathbb{R}^D$, how far is $\mathbf{x}$ from H?



$$w \cdot x = \|w\|_2 \, \|proj_w x\|_2$$

$$\|proj_w x\|_2 = \frac{w \cdot x}{\|w\|_2}$$

$$\gamma = \|proj_w x\|_2 - \|proj_w y\|_2.$$

$$w \cdot y = \|w\|_2 \, \|proj_w y\|_2$$

$$w \cdot y + w_0 = 0 \Rightarrow w \cdot y = -w_0$$

$$\|proj_w y\|_2 = -\frac{w_0}{\|w\|_2}$$

$$\gamma = \frac{w \cdot x}{\|w\|_2} - \left(-\frac{w_0}{\|w\|_2}\right)$$

$$= \frac{w \cdot x + w_0}{\|w\|_2} \checkmark.$$

Distance of Point to Hyperplane

- Consider the hyperplane: $H \equiv \mathbf{w} \cdot \mathbf{x} + w_0 = 0$
- All points on the positive halfspace created by H satisfy

$$\mathbf{w} \cdot \mathbf{x} + w_0 > 0$$

  All points on the negative halfspace created by H satisfy

$$\mathbf{w} \cdot \mathbf{x} + w_0 < 0$$

- We label $y = +1$ ($y = -1$) the data points in the positive (negative) halfspace
- The distance of point $\mathbf{x}$ to H is given by:

$$\frac{|\mathbf{w} \cdot \mathbf{x} + w_0|}{\|\mathbf{w}\|_2}$$

Alternative Formulation as Optimisation Problem

$\boxed{\text{Find distance } \|\mathbf{x} - \mathbf{x}^*\|_2 \text{ between point } \mathbf{x}^* \text{ and the hyperplane } \mathbf{w} \cdot \mathbf{x} + w_0 = 0}$

The point $\mathbf{x}$ on hyperplane that is closest to $\mathbf{x}^*$ gives the distance

Equivalently, we seek for $\mathbf{x}$ that optimises the following problem:

$$\begin{aligned} \text{minimise:} \quad & \|\mathbf{x} - \mathbf{x}^*\|_2^2 \\ \text{subject to:} \quad & \mathbf{w} \cdot \mathbf{x} + w_0 = 0 \end{aligned}$$

Lagrangian:

$$\begin{aligned} \Lambda(\mathbf{x}, \lambda) &= \|\mathbf{x} - \mathbf{x}^*\|_2^2 - 2\lambda(\mathbf{w} \cdot \mathbf{x} + w_0) \\ &= \|\mathbf{x}\|_2^2 - 2(\mathbf{x}^* + \lambda\mathbf{w}) \cdot \mathbf{x} - 2\lambda w_0 + \|\mathbf{x}^*\|_2^2 \end{aligned}$$

Find critical point of Λ by setting its gradient to $\mathbf{0}$:

$$\nabla_{\mathbf{x}}\Lambda(\mathbf{x}, \lambda) = 2\mathbf{x} - 2\mathbf{x}^* - 2\lambda\mathbf{w} = 0 \Rightarrow \mathbf{x} = \mathbf{x}^* + \lambda\mathbf{w}$$

Alternative Formulation as Optimisation Problem – Continued

From previous slide: We obtained the critical point $\mathbf{x} = \mathbf{x}^* + \lambda\mathbf{w}$

We next obtain an expression for λ by substituting $\mathbf{x}$ into the hyperplane equation:

$$\mathbf{w} \cdot (\mathbf{x}^* + \lambda\mathbf{w}) + w_0 = 0 \Rightarrow \lambda = -\frac{\mathbf{w} \cdot \mathbf{x}^* + w_0}{\mathbf{w} \cdot \mathbf{w}} = -\frac{\mathbf{w} \cdot \mathbf{x}^* + w_0}{\|\mathbf{w}\|_2^2}$$

Finally, the distance between $\mathbf{x}^*$ and $\mathbf{x}$ becomes:

$$\|\mathbf{x} - \mathbf{x}^*\|_2 = \|\lambda\mathbf{w}\|_2 = |\lambda| \, \|\mathbf{w}\|_2 = \frac{|\mathbf{w} \cdot \mathbf{x}^* + w_0|}{\|\mathbf{w}\|_2}$$

The Linearly Separable Case

Assume that the dataset $\mathcal{D} = \big((\mathbf{x}_i, y_i), \ldots, (\mathbf{x}_N, y_N)\big)$ is linearly separable

Recall

$\mathbf{w} \cdot \mathbf{x}_i + w_0 > 0$ and $y_i = +1$ or $\mathbf{w} \cdot \mathbf{x}_i + w_0 < 0$ and $y_i = -1$

More compactly,

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) > 0$$

Since $\mathcal{D}$ is finite, we can always find $\epsilon > 0$ such that

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) \geq \epsilon$$

Alternative formulation (divide both sides by ϵ):

$$y_i(\underbrace{\frac{\mathbf{w}}{\epsilon}}_{\text{new } \mathbf{w}} \cdot \mathbf{x}_i + \underbrace{\frac{w_0}{\epsilon}}_{\text{new } w_0}) \geq 1$$

Margin Maximisation

Recall the margin of data point $\mathbf{x}_i$ to hyperplane:

$$\frac{|\mathbf{w} \cdot \mathbf{x}_i + w_0|}{\|\mathbf{w}\|_2} = \frac{y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0)}{\|\mathbf{w}\|_2} \geq \frac{1}{\|\mathbf{w}\|_2}$$

Maximum margin principle: pick the boundary that maximises the margin

$$\begin{aligned} \text{maximise:} \quad & \frac{1}{\|\mathbf{w}\|_2} \\ \text{subject to:} \quad & y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) \geq 1, \text{ for } 1 \leq i \leq N \end{aligned}$$

Equivalently, we minimise the squared ℓ_2 norm of $\mathbf{w}$ subject to the constraints

$$\begin{aligned} \text{minimise:} \quad & \tfrac{1}{2}\|\mathbf{w}\|_2^2 \\ \text{subject to:} \quad & y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) \geq 1, \text{ for } 1 \leq i \leq N \end{aligned}$$

The constant factor $\frac{1}{2}$ is added without loss of generality.

Margin Maximisation is a Convex Quadratic Optimisation Problem

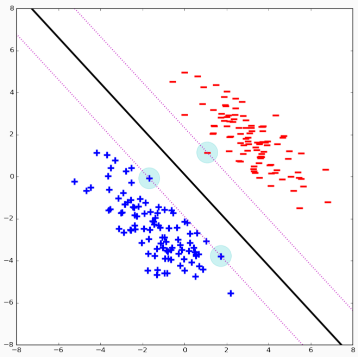
$$\begin{aligned} \text{minimise:} \quad & \tfrac{1}{2}\|\mathbf{w}\|_2^2 \\ \text{subject to:} \quad & y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) \geq 1, \text{ for } 1 \leq i \leq N \end{aligned}$$

- The objective is a convex quadratic function
- The constraints are convex linear functions
- The feasible set as defined by the constraints is convex as well

$\Rightarrow$ Our optimisation problem is convex quadratic

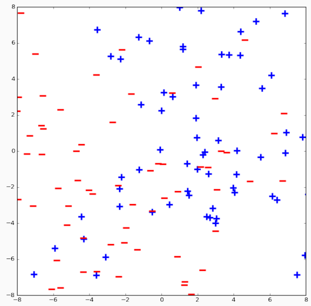
$\Rightarrow$ Solvable using generic convex optimisation methods

SVM in the Linearly Separable Case



We find an SVM classifier with <u>no classification error</u> on the training set

The Non-Separable Case



- The quadratic program from the previous slides has no feasible solution
- What would be a good relaxation of the previous optimisation problem?

Relaxation of the SVM Formulation

Original SVM formulation for linearly-separable data:

$$\begin{aligned} \text{minimise:} \quad & \tfrac{1}{2}\|\mathbf{w}\|_2^2 \\ \text{subject to:} \quad & y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) \geq 1 \qquad \text{for } 1 \leq i \leq N \end{aligned}$$

Relaxations:

1. Find a separator that makes the least mistakes on the training error
   - Minimising the number of misclassifications is NP-hard :(
2. Proxy for least mistakes: Satisfy as many of the N constraints as possible
3. Alternative: Allow all constraints to be satisfied with some slack

$$\begin{aligned} \text{minimise:} \quad & \tfrac{1}{2}\|\mathbf{w}\|_2^2 + C\textstyle\sum_{i=1}^{N}\zeta_i \\ \text{subject to:} \quad & y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) \geq 1 - \zeta_i \quad \text{for } 1 \leq i \leq N \\ & \zeta_i \geq 0 \qquad\qquad\qquad\quad\ \text{for } 1 \leq i \leq N \end{aligned}$$

SVM Formulation in the Non-Separable Case

$$\begin{aligned} \text{minimise:} \quad & \tfrac{1}{2}\|\mathbf{w}\|_2^2 + C\textstyle\sum_{i=1}^{N}\zeta_i \\ \text{subject to:} \quad & y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) \geq 1 - \zeta_i \quad \text{for } 1 \leq i \leq N \\ & \zeta_i \geq 0 \qquad\qquad\qquad\quad\ \text{for } 1 \leq i \leq N \end{aligned}$$

- Feasible solution always exists thanks to slack variables ζ_i
  - ζ_i can be as large (constraints can be violated as much) as necessary
- The constraints $\zeta_i \geq 0$ are important!
  - Assume $\zeta_i \ll 0$ and $\mathbf{x}_i$ correctly classified by a huge margin
  - This would compensate the penalty incurred on misclassified points
  - $\zeta_i \geq 0$ ensures no bonus for correct classification by a huge margin

SVM Formulation: Loss Function

$$\begin{aligned} \text{minimise:} \quad & \tfrac{1}{2}\|\mathbf{w}\|_2^2 + C\textstyle\sum_{i=1}^{N}\zeta_i \\ \text{subject to:} \quad & y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) \geq 1 - \zeta_i \quad \text{for } 1 \leq i \leq N \\ & \zeta_i \geq 0 \qquad\qquad\qquad\quad\ \text{for } 1 \leq i \leq N \end{aligned}$$

Optimal solution must satisfy

- either $\zeta_i = 0$: This is ideal, no slack is needed to classify $\mathbf{x}_i$
- or $\zeta_i = 1 - y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) \geq 0$: Minimum ζ_i that satisfies the constraint

Equivalent formulation of the optimal solution for ζ_i:

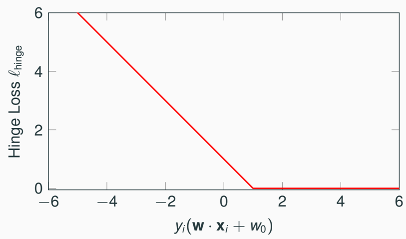
$$\zeta_i = \max(0, 1 - y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0)) \stackrel{def}{=} \ell_{\text{hinge}}(\mathbf{w}, w_0; \mathbf{x}_i, y_i)$$

Our SVM formulation becomes equivalent to minimising the objective function:

$$\mathcal{L}_{\text{SVM}}(\mathbf{w}, w_0 \mid \mathbf{X}, \mathbf{y}) = \underbrace{\frac{1}{2}\|\mathbf{w}\|_2^2}_{\ell_2 \text{ regulariser}} + \underbrace{C\sum_{i=1}^{N}\ell_{\text{hinge}}(\mathbf{w}, w_0; \mathbf{x}_i, y_i)}_{\text{hinge loss}}$$

Hinge Loss Function

$$\ell_{\text{hinge}}(\mathbf{w}, w_0; \mathbf{x}_i, y_i) = \max(0, 1 - y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0))$$



Logistic Loss Function

$y_i \in \{0, 1\}$ for logistic regression and $y_i \in \{-1, +1\}$ for SVM

We resolve the mismatch by using $z_i = 2y_i - 1$ to map from $\{0, 1\}$ to $\{-1, +1\}$.

$$\mathrm{NLL}(y_i \mid \mathbf{w}, \mathbf{x}_i) = -\left(y_i \log \left(\frac{1}{1 + e^{-\mathbf{w} \cdot \mathbf{x}_i}} \right) + (1 - y_i) \log \left(\frac{1}{1 + e^{\mathbf{w} \cdot \mathbf{x}_i}} \right) \right)$$

If $y_i = 1$, then $z_i = 1$ and:

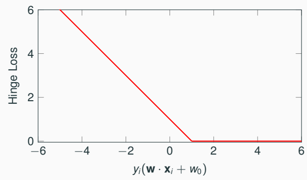
$$\mathrm{NLL}(y_i = 1 \mid \mathbf{w}, \mathbf{x}_i) = -\log \left(\frac{1}{1 + e^{-\mathbf{w} \cdot \mathbf{x}_i}} \right) = \log \left(1 + e^{-\mathbf{w} \cdot \mathbf{x}_i} \right) = \log \left(1 + e^{-z_i \mathbf{w} \cdot \mathbf{x}_i} \right)$$

If $y_i = 0$, then $z_i = -1$ and obtain the logistic loss function:

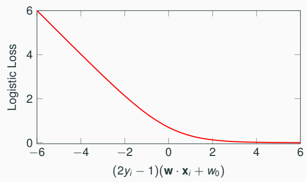
$$\mathrm{NLL}(y_i = 0 \mid \mathbf{w}, \mathbf{x}_i) = -\log \left(\frac{1}{1 + e^{\mathbf{w} \cdot \mathbf{x}_i}} \right) = \log \left(1 + e^{\mathbf{w} \cdot \mathbf{x}_i} \right) = \log \left(1 + e^{-z_i \mathbf{w} \cdot \mathbf{x}_i} \right)$$

By replacing $z_i = (2y_i - 1)$, we obtain: $\mathrm{NLL}(y_i \mid \mathbf{w}, \mathbf{x}_i) = \log \left(1 + e^{-(2y_i - 1) \mathbf{w} \cdot \mathbf{x}_i} \right)$

Loss Functions: Hinge vs Logistic



$\max(0, 1 - y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0))$



$\log \left(1 + e^{-(2y_i - 1)(\mathbf{w} \cdot \mathbf{x}_i + w_0)} \right)$