

Support Vector Machine (SVM)

1. Was ist SVM? Wofür benutzt und ist es supervised oder unsupervised?

- Maschinelles Lernverfahren.
 - Klassifikation und Regression
 - Supervised (Label müssen vorhanden sein)

2. Was ist das Ziel von SVM?

- Finde optimale Trennebene / Hyperebene für Klassen.

3. Seit wann existiert SVM als Theorie und seit wann wird es benutzt?

- Idee: Fisher 1936
- Einsatz als ML-Methode: Boser et al. 1992

4. Was ist eine Hyperebene?

Mathematische Konstruktion (geometrische Form), die zur Trennung von Daten in einem mehr dimensional Raum verwendet wird.

- Optimale Ebene, die Klassen auftrennt.

5. Welche Form und wie viele Dimensionen hat die Hyperebene?

- Eine Hyperebene ist in der linearen Algebra und Geometrie ein Begriff, der sich auf einen Unterraum der Dimension $n-1$ in einem n -dimensionalen Vektorraum bezieht.
- Es hat also immer $n-1$ Dimensionen.

Bsp.

- 2D → Linie (1D)
- 3D → Ebene (2D)
- 4D → Quader (3D).
- nD → Hyperquader mit $n-1$ D.

6. Was ist die optimale Hyperebene ? – Hyperebenenkriterium kennen

- Minimaler Abstand aller Stützvektoren zur optimalen Hyperebene ist maximal.
- Dh. Abstand so wählen, dass Margin maximiert wird.

7. Welche Punkte werden primär genutzt für das Training der SVM?

- Die Grenzpunkte
- Optimale Ebene nur von ihnen abhängig

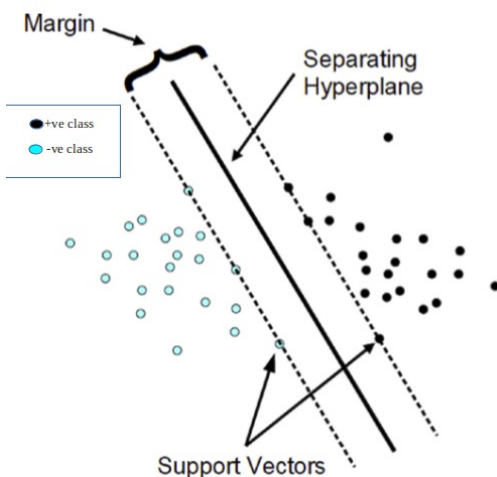
8. Was ist ein anderer Name für diese Punkte + definiere die Punkte? Wann gültig?

- Stützvektoren / Support Vector
- Definieren Hyperebene: (Support Vektoren (auch als Stützvektoren oder Supportpunkte bezeichnet) sind diejenigen Datenpunkte in einem Datensatz, die die Position und Orientierung der Entscheidungsebene oder Hyperplane in einem Support Vector Machine (SVM)-Modell maßgeblich beeinflussen.)
- Gültigkeit: Ein einzelner Support Vektor kann tatsächlich nur eine einzige Hyperebene definieren.

9. Wieso heißt es SVM?

- Weil es von Grenzpunkten/Supportvektoren abhängig ist, wo Hyperebene platziert wird und wie es platziert wird.

10. Was ist eine Margin?



→ Abstand zwischen Hyperebene und Grenzpunkten/ Stützvektoren.

11. Welchen anderen Namen hat SVM und warum?

- Maximum Margin Classifier.
- Wegen dem Optimierungskriterium, wonach die optimale Hyperebene den Margin maximiert.

12. Was ist das Ergebnis des Trainings?

- Funktion, die Hyperebene definiert.
- Linearkombination von Stützvektoren.

13. Wie weist SVM (neue) Punkte Klassen zu?

- Lage des Punktes relativ zur Hyperebene wird betrachtet (oberhalb oder unterhalb der Hyperebene?)

14. Wann sind Daten linear separierbar und wann nicht?

Daten sind "linear separierbar", wenn du eine gerade Linie (in zwei Dimensionen), eine Ebene (in drei Dimensionen) oder eine Hyperebene (in mehr als drei Dimensionen) finden kannst, die die Daten in zwei Gruppen aufteilt.

Daten sind "nicht linear separierbar", wenn es keine einfache gerade Linie, Ebene oder Hyperebene gibt, um die Daten zu trennen. Dies passiert oft, wenn die Datenpunkte beider Klassen miteinander vermischt oder überlappend sind

15. Was ist der Kerneltrick und wann wird er benutzt?

- Bei nicht linear separierbaren Daten benutzt.
- Definition: Eine nicht lineare Operation, mit der man nicht linear separierbare Daten in eine höhere Dimension (Feature Raum) transferiert, wo Sie linear separierbar sind.
- Benutzt Skalare, nicht die echten Daten.

16. Gebe 3 Beispiele für einen Kernel und wo es verwendet wird.

- Sigmoid: Neuronale Netze
- Gauss-Kernel: Bildverarbeitung (Kantenglättung, siehe Bildanalyse VL).
- Laplace: Bildverarbeitung (Kantendetektion, siehe Bildanalyse VL)

17. Wie nutzt man SVM bei Multiklassen-Problemen?

- 2 Ansätze: Beide Ansätze setzen voraus, dass man mehrere SVMs trainiert.

1. One vs All

- Trenne Daten binär auf (jeweils 1 Klasse gegen alle Klassen zusammengefasst)
- 1 SVM für jede Klasse trainieren: bei 4 Klassen → 4 mal trainieren.
- Winner takes all Prinzip: Der Datenpunkt wird der Klasse zugeordnet, für die die SVM den höchsten Wahrscheinlichkeitswert oder Vertrauensscore berechnet hat.

2. One vs One

- Je Klassenpaar wird 1 SVM trainiert.
- Brauche mehr SVMs: Höhere Rechenzeit
- Max-wins-votings

18. Welche Gleichung für Hyperebene benutzt?

Lineare Gleichung

- 2D: $y = w_1x + b$
- 3D: $z = w_1x + w_2y + b$
- nD: $H = w^T x + b$

$$\bullet w = \begin{pmatrix} w_1 \\ w_2 \\ \dots \\ w_n \end{pmatrix}$$

$$\bullet x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$$

$$\rightarrow H = w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_nx_n + b$$

19. Worüber Klassenzugehörigkeit bestimmt.

$$class(x) = sign(w * x - b)$$

→ Klasse über Vorzeichen (Sign) bestimmt.

- Relative Lage zur Hyperebene durch Vorzeichen definiert.
- Entweder -1 oder +1

19. Welche Aufgabe hat der Error-Term?

- Klasse 2: $w * x - b \leq 1 - \varepsilon$
- $\varepsilon \geq 0$

→ Overfitting verhindern

→ Ausreißer erlauben (Ausreißer auf falscher Seite der Hyperebene)

20. Wie wird trainiert? Wie wird maximaler Margin definiert?

$$d_H(x) = \frac{|w * x - b|}{\|w\|^2}$$

Distanz d zur
Hyperebene H

$$w^* = \operatorname{argmax}_w (\min_n d_H(x_n))$$

Optimierungsproblem
 w^* : optimaler Abstand

$$w^* = \operatorname{argmax}_w (\min_n \frac{|w * x_n - b|}{\|w\|^2})$$

Einsetzen

$$\min \frac{1}{2} \|w\|^2$$

Formulierung
Lagrange

in Abhängigkeit von

$$y_i (|w * x_i - b|) \geq 1, \forall i$$



21. Wozu dient Lagrange Funktion beim Training?

In der Kontext der Support Vector Machines (SVM) wird die Lagrange-Funktion im Rahmen der Optimierungsaufgabe verwendet, um die Parameter der Entscheidungsebene zu optimieren. SVMs suchen nach einer optimalen Entscheidungsebene, die den Raum in zwei Klassen aufteilt und gleichzeitig den sogenannten Margin maximiert.

Die Lagrange-Funktion, auch als Lagrange-Funktional oder Lagrange-Formalismus bekannt, wird in diesem Zusammenhang verwendet, **um die Nebenbedingungen der Optimierungsaufgabe zu berücksichtigen**. In einfachen Worten helfen die Lagrange-Multiplikatoren, die Optimierungsaufgabe unter bestimmten Bedingungen zu lösen.

Nebenbedingung: Dazu gehört, dass alle Datenpunkte auf der richtigen Seite der Entscheidungsebene (der Seite, die ihrer Klasse entspricht) liegen und der Abstand zu den Support Vektoren maximal ist.