Technische Universität Berlin **SoSe 2022**
Fakultät II, Institut für Mathematik
Sekretariat MA 5–2, Dorothea Kiefer-Hoeft
Prof. Dr. Martin Skutella
Dr. Frank Lutz, Ekin Ergen, Martin Knaack, Manuel Radons

Programmierprojekt Computerorientierte Mathematik II

Vorstellung: bis **12.7.2022** bei Ihrer Tutorin oder Ihrem Tutor

Das Programmierprojekt ist in Gruppen von höchstens drei Student:innen zu bearbeiten. Die Vorstellung ist bis einschließlich dem 12.7.2022 möglich.

Einführung

Ziel des Programmierprojekts ist es, einen Klassifikator für Punktmengen, eine sogenannte Support-Vector Machine (hiernach SVM; eine deutsche Übersetzung des Begriffs ist unüblich), partiell zu implementieren, zu testen und die Ergebnisse der Tests zu visualisieren.
Zwecks Einfachheit der Darstellung sowie der Durchführung beschränken wir uns auf den 2-dimensionalen Fall. Es sei eine Menge von 2-Tupeln der Form (x_i, y_i) gegeben, wobei $x_i \in \mathbb{R}^2$, $y_i \in \{-1, 1\}$ und $i = 1, 2, \ldots, k$. Wir nehmen an, dass die Punkte x_i so verteilt liegen, dass es möglich ist, eine affine Gerade so in die Ebene zu legen, dass in einer der dadurch ausgezeichneten Halbebenen nur Punkte x_i liegen, so dass $y_i = -1$ und in der anderen Halbebene nur solche Punkte x_i so dass $y_i = 1$. Eine SVM berechnet eine solche Gerade, welche überdies den Abstand zu beiden Punktmengen maximiert, siehe Abbildung 1.
Es ist möglich das Konzept der SVM mittels des sogenannten Kernel-Tricks auch auf solche Punktmengen zu erweitern, die sich nicht durch eine affine Gerade trennen lassen. Aber dies wird nicht Teil der Aufgabenstellung sein.

Mathematische Formulierung

Eine Gerade im $\mathbb{R}^2$ genügt der Gleichung

$$w^\top x - b \;=\; \langle w, x\rangle - b \;=\; 0\,,$$

wobei $w \in \mathbb{R}^2$ ein Normalenvektor zur Geraden und $b \in \mathbb{R}$ der (in unserem Kontext) sogenannte Bias sind. Mit $\langle\cdot,\cdot\rangle$ wird das standard Skalarprodukt im Euklidischen Raum bezeichnet.
Für Punkte, die obige Gleichung nicht erfüllen, gibt das Vorzeichen des Ausdrucks $w^\top x - b$ die Seite der Gerade an, auf der x liegt. Eine die Punktmengen separierende Gerade ist also eine Gerade mit Parametern w und b so dass $w^\top x_i - b$ größer/kleiner null ist genau dann wenn $y_i = 1$ bzw. $y_i = -1$.
Wie oben erwäht, wollen wir eine besondere separierende Gerade finden, nämlich jene, die den maximalen Abstand zu beiden Punktmengen hat. Diese Bedingung lässt sich forcieren durch die Ungleichungen

$$w^\top x - b \;\geq 1\,,\ \text{falls } y_i = 1 \quad \text{und} \quad w^\top x - b \;\leq 1\,,\ \text{falls } y_i = -1\,,$$

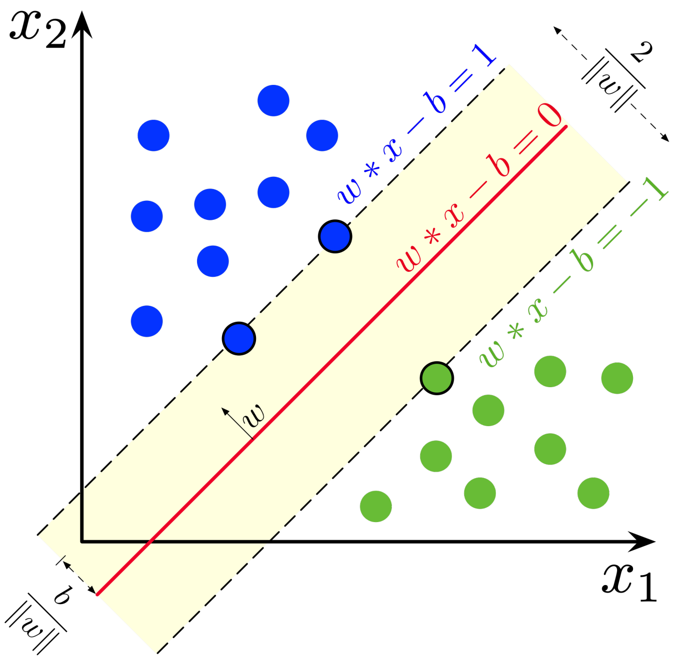

Abbildung 1: Quelle: Wikipedia

welche sich zusammenfassen lassen in der Ungleichung

$$y_i(w^\top x - b) \;\geq 1 \quad \text{für alle } i \in \{1, 2, \ldots, k\}\,. \tag{1}$$

Hieraus ergibt sich das Optimierungsproblem

$$\min_w \|w\|$$
$$\text{so dass (1)}$$

In Worten: Ermittle ein $w \in \mathbb{R}^n$ mit minimaler Norm, so dass (1) erfüllt ist.

Formulierung als quadratisches Programm

Das Problem, die bestmögliche Gerade zu finden, läßt sich als sogenanntes quadratisches Programm (hiernach QP) formulieren. Sie müssen sich hierbei nicht von der unten stehenden Formulierung verunsichern lassen. Für die Lösung des Problems reicht es aus, nach dem angegebenen Bauprinzip die Matrizen aufzustellen und dem vorgeschlagenen QP Löser (oder einem anderen QP Löser Ihrer Wahl in `Julia`) zu übergeben.
Zur Aufstellung des quadratischen Programms konstruieren wir die Matrix $M \in \mathbb{R}^{k \times k}$ wiefolgt aus unserer Tupelmenge. Es sei

$$M_{ij} \;=\; y_i y_j \langle x_i, x_j \rangle\,.$$

Es bezeichne nun **1** den Vektor aus 1-en. Dann gilt es, folgendes Optimierungsproblem zu lösen, welches aufgrund der beiderseitigen Multiplikation von M mit einem Vektor $\alpha \in \mathbb{R}^k$ auch quadratisches Programm genannt wird:

$$\begin{aligned} \min_{\alpha} \; & \tfrac{1}{2}\,\alpha^\top M\alpha \;-\; \mathbf{1}^\top\alpha \\ \text{so dass } \sum_{i=1}^{k} \alpha_i y_i \; & = \; 0 \\ \text{und } -\alpha_i \; & \leq \; 0 \;\forall\; i \in \{1,2,\dots,k\} \end{aligned}$$

In Worten: Minimiere den Ausdruck in der ersten Zeile bezüglich α so dass α die Nebenbedingungen in der zweiten und dritten Zeile erfüllt.
Haben wir α ermittelt, ergibt sich w wiefolgt:

$$w \;=\; \sum_{i=1}^{k} \alpha_i y_i x_i\,.$$

Es sei nun $\mathcal{I}$ die Menge aller Indizes in $\{1,2,\dots,k\}$ so dass $\alpha_i > 10^{-5}$ und es sei I die Zahl der Elemente in $\mathcal{I}$. (Es sei hierbei erwähnt, dass der Ausdruck $\alpha_i > 10^{-5}$ eine numerische Approximation für den exakten Ausdruck $\alpha > 0$ ist.) Dann gilt

$$b = \frac{1}{I}\sum_{i\in\mathcal{I}}\left(y_i - \sum_{j\in\mathcal{I}} \alpha_j y_j \boldsymbol{x}_j^\top \boldsymbol{x}_i\right)$$

Lösung in `Julia` mittels `QPDAS`

Eine Möglichkeit, die α_i zu berechnen, ist mittels des `Julia` Pakets `QPDAS`. Die Website des Pakets ist

`https://juliahub.com/ui/Packages/QPDAS/9vHhA/0.2.0`

Das Paket kann installiert werden mittels des Befehls

Pkg.add("QPDAS")

Der Löser in dem Paket erwartet ein quadratisches Programm der Form

$$\begin{aligned} \min_{x} \; & \tfrac{1}{2}\,x^\top P x \;-\; z^\top x \\ \text{so dass } Ax \; & = \; b \\ \text{und } Cx \; & \leq \; d \end{aligned}$$

Dabei sind A, C Matrizen, b, d Vektoren und $=$, $\leq$ komponentenweise Vergleiche. (Überlegen Sie sich, wie A, C, b, d beschaffen sein müssen, um die oben formulierten Bedingungen zu realisieren.)
Sie sind frei darin, andere Pakete zu nutzen, solange es sich dabei um `Julia` Pakete handelt

Anforderungen

Ein erfolgreich vorgestelltes Projekt muss den folgenden Mindestanforderungen genügen:

1. *Daten einlesen.* Stellen Sie eine Methode in `Julia` bereit, welche Textdateien mit Beispielproblemen einliest und eine Menge von Tupeln der oben beschriebenen Form (x_i, y_i) erzeugt. Das Format der Daten ist das Folgende: In jeder Zeile der einzulesenden Datei befinden sich drei Zahlen. Die ersten zwei sind Floating Points mit den Koordinaten von x_i, die letzte Zahl ist -1 oder $+1$.

2. *Quadratisches Programm.* Stellen Sie das oben vorgestellte quadratische Programm auf in einer für den Löser Ihrer wahl geeigneten Form.

3. *Berechnungen.* Berechnen Sie w und b für verschiedene Tupelmengen. Eine absolute Mindestanforderung ist das Bearbeiten der beiden vorgegebenen, sowie zweier eigener Beispiele.

4. *Visualisierung.* Visualisieren Sie Ihre Ergebnisse mit der `Python` Bibliothek `Matplotlib`. Teil dieser Aufgabe ist ist die Übergabe der Ergebnisse Ihrer Berechnungen an die `Python` Bibliothek. Sie sind frei darin, wie Sie diese Übergabe gestalten.

Sie sind natürlich frei, hierüber hinaus zu gehen, zum Beispiel durch mehr Beispiele, höherdimensionale Beispiele, oder Beispiele von nicht durch eine gerade trennbare Punktmengen und Trennung derselben mittels des Kernel Tricks.