9.5 Entscheidungsbäume

Wir betrachten wieder einen Datensatz von Ereignissen mit jeweils m Merkmalen, zusammengefasst in $\vec{x}$, die zwei verschiedenen Klassen angehören, zum Beispiel 'Signal' und 'Untergrund'. Im folgenden soll die Klassifizierung durch Entscheidungsbäume ('decision trees') eingeführt werden: Sequentielle Anwendung von Trennschnitten auf die Merkmale der Ereignisse verteilt die Daten auf verschiedene Äste, an deren Enden jeweils ein Blatt einer bestimmten Klasse zugeordnet ist. Zu derselben Klasse kann es mehrere Blätter geben, aber jedes Blatt ist nur auf einem Weg zu erreichen.

Im binären Entscheidungsbaum wird eine Serie von Fragen gestellt, welche alle mit Ja oder Nein beantwortet werden können. Diese Serie ergibt ein Resultat, welches durch eine Regel bestimmt ist. Die Regel ist einfach ablesbar, wenn man von der Wurzel her den Ästen des Baumes folgt, bis man zu einem bestimmten Blatt gelangt, welches das Resultat der Fragereihe darstellt. Bei kontinuierlich verteilten Merkmalen werden Trennschnitte bestimmt, die dann letztlich an jedem Punkt zu einer Ja-Nein-Entscheidung führt (größer/kleiner).

Generiert werden die Entscheidungsbäume üblicherweise bei der Wurzel beginnend schrittweise bis zu den Blättern gehend. Bei jedem Schritt wird genau das Merkmal gesucht, mit welchem man die Daten am besten klassifizieren kann. Um dieses zu ermitteln, muss die beste Aufteilung gefunden werden, das heißt die Aufteilung der Daten muss so gewählt werden, dass sie nach der Aufteilung möglichst rein sind. Ein Maß für die Reinheit ist zum Beispiel die Entropie. Aus der Entropie lässt sich dann berechnen, welches Merkmal für die Verzweigung den höchsten Informationsgewinn bietet. Ein weiteres Maß für die Bestimmung der optimalen Aufteilung ist der Gini-Index, der im Folgenden benutzt wird.

9.5.1 Aufwachsen eines Baumes

Ein Entscheidungsbaum wird mit einem Trainingsdatensatz in folgenden Schritten konstruiert:

1. Beginne mit dem Trainingsdatensatz an der Wurzel ('root node').

2. Suche aus allen Ereignissen das signifikanteste Merkmal zum Aufteilen an diesem Knoten.

3. Teile nach einem Kriterium wie zum Beispiel dem maximalen Gini-Index[1] auf:

$$G = 4\,P(1-P) \tag{9.41}$$

[1]Der Gini-Index ist ein Maß für Ungleichverteilung. Die hier benutzte Definition weicht etwas von der üblichen ab, bei der $G = (A_{gleich} - A)/A_{gleich}$ ist, wobei A_{gleich} die Fläche unter der Verteilungsfunktion für eine Gleichverteilung und A die Fläche unter der tatsächlichen Verteilungsfunktion ist. Aus Wikipedia: "Der Gini-Index oder auch Gini-Koeffizient ist ein statistisches Maß, das vom italienischen Statistiker Corrado Gini zur Darstellung von Ungleichverteilungen entwickelt wurde. Der Koeffizient kann beispielsweise als Kennzahl für die Ungleichverteilung von Einkommen oder Vermögen eingesetzt werden. Er wird besonders in der Wohlfahrtsökonomie verwendet."

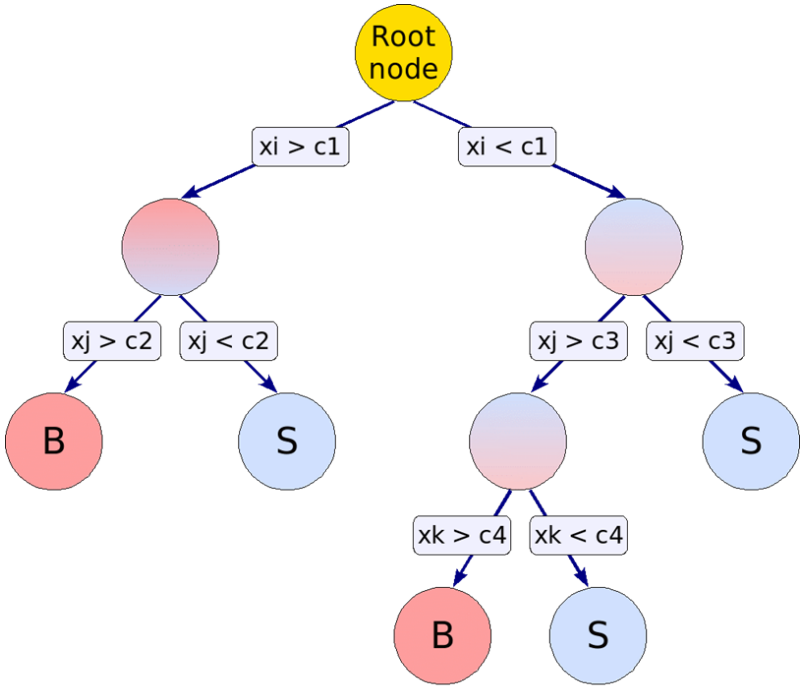

Abbildung 9.30: Prinzip eines Entscheidungsbaumes: an jeder Verzweigung (Knoten) wird das Merkmal ausgesucht, das die signifikanteste Trennung durch einen Selektionsschnitt erlaubt. Die Blätter (Endknoten) sind einer Klasse zugeordnet (hier S=Signal und B=Untergrund).

Dabei ist P die 'Reinheit' der Klasse 1, die bei einem Schnitt auf ein Merkmal an einem Knoten erreicht wird, und $1 - P$ die Reinheit der Klasse 2:

$$P = \frac{N_1}{N_1 + N_2} \tag{9.42}$$

Der Gini-Index wird für $P = 0.5$, entsprechend $G = 1$, maximal (mit der Normierung in (9.41) gilt $0 \leq G \leq 1$).

Der Gini-Index wird für die Bestimmung des Merkmals, das an einem Knoten die signifikanteste Trennung bietet und für die Bestimmung des Trennschnitts benutzt (maximiert).

4. Setze die Aufteilung fort, bis ein vorgegebenes Abbruchkriterium erfüllt ist, bis zum Beispiel eine minimale Anzahl Ereignisse in einem Knoten verbleibt oder bis eine maximale Reinheit erreicht ist.

5. Ein Blatt wird der Klasse zugeordnet, die die meisten Ereignisse in dem Blatt hat.

6. Evaluiere Effizienz und Reinheit mit einem unabhängigen und dem Baum bisher unbekannten Testdatensatz.

Für die Klassifizierung von Daten und die Lösung von Fragestellungen auf der Basis von Daten werden in den unterschiedlichsten Bereichen (Wirtschaft, Medizin,

Naturwissenschaften, ...) häufig Entscheidungsbäume benutzt. Die vorteilhaften Eigenschaften sind:

- Unabhängigkeit von gleichförmigen Variablentransformationen;
- Unanfälligkeit gegen Ausreißer in den Daten;
- Unterdrückung von 'schwachen' Variablen ohne Verlust der Leistungsfähigkeit.

Schwachstellen sind:

- Instabilität der Baumstruktur gegenüber kleinen Änderungen der Trainingsdaten;
- Anfälligkeit auf Übertraining (Abhilfe: 'pruning' = 'Ausasten');

Eine Klassifizierung mit einem Entscheidungsbaum hat also einige nicht ganz optimale Eigenschaften. Eine wesentliche Verbesserung stellen 'verstärkte Entscheidungsbäume' dar, wie im Folgenden besprochen wird.

9.5.2 Verstärkte Entscheidungsbäume

Ein weitaus besseres Klassifikationsvermögen wird dadurch erreicht, dass viele Bäume generiert werden und deren Ergebnisse gemittelt werden. Nach jeder Erzeugung eines Baumes gehen die falschen Zuordnungen mit einem höheren Gewicht in die nächste Erzeugung eines Baumes ein, wodurch sie mit höherer Wahrscheinlichkeit richtig eingeordnet werden. Die Klassenzugehörigkeit wird durch Mittelung der Entscheidung aller Bäume ermittelt ('verstärkte Entscheidungsbäume', 'boosted decision trees').

Das Training beginnt wie bei einem einzelnen Baum, wobei alle Ereignisse das Gewicht 1 haben. Bei der Erzeugung des nächsten Baumes wird jedem Ereignis ein Gewicht w_i zugeordnet, das von dem angewandten Algorithmus abhängt. Die Berechnung der Reinheit P in (9.42) ändert sich dann entsprechend zu

$$P = \frac{\sum_{i=1}^{N_1} w_i}{\sum_{i=1}^{N_1} w_i \; + \; \sum_{i=1}^{N_2} w_i} \tag{9.43}$$

Nach der Fertigstellung des Baumes werden die Gewichte wieder für den nächsten Baum berechnet. Das geht so weiter bis eine vorggebene Maximalzahl M von Bäumen generiert worden ist (typisch $M \approx 1000$).

Die Entscheidungsfunktion eines einzelnen Baumes sei:

$$y_k(\vec{x}) = \pm 1, \qquad k = 1, \ldots, M, \tag{9.44}$$

(zum Beispiel $y = +1$ für Klasse 1 und $y = -1$ für Klasse 2). Für die Gesamtentscheidung wird das gewichtete Mittel der einzelnen Entscheidungen gebildet:

$$y(\vec{x}) = \frac{\sum_{k=1}^{M} g_k \, y_k(\vec{x})}{\sum_{k=1}^{M} g_k} \tag{9.45}$$

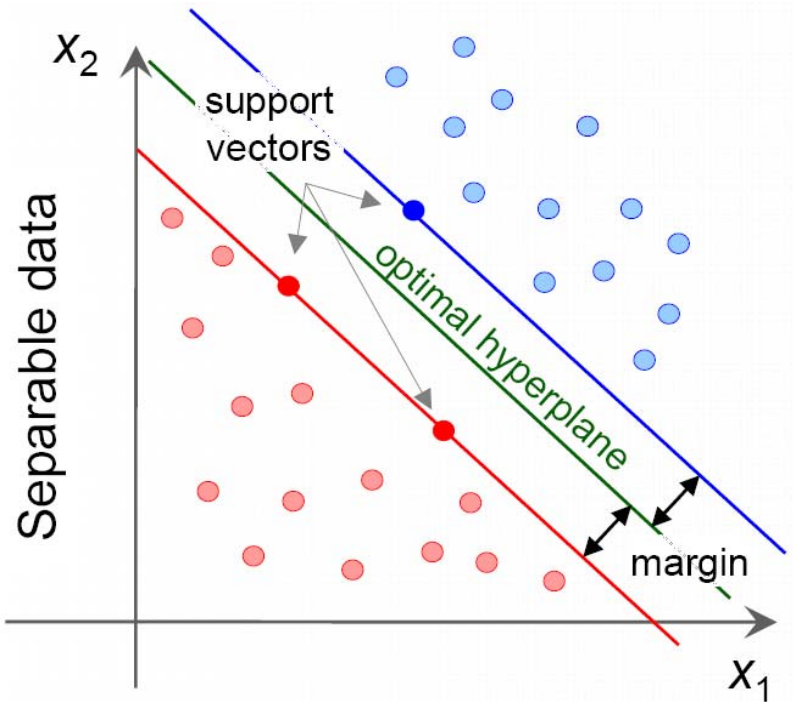

Abbildung 9.31: Zwei disjunkte Datenmengen, die durch eine Diskriminante mit der größten Trennspanne separiert werden.

Die Gewichte werden so gewählt, dass eine dem speziellen Algorithmus zugeordnete Verlustfunktion, die im Allgemeinen eine Funktion der richtigen und falschen Zuordnungen ist, minimiert wird. Als Beispiel ist der Algorithmus AdaBoost in [4] erklärt. In der TeV-Gamma-Astronomie (MAGIC, HESS) ist 'Random Forest'[2] beliebt.

Mit 'boosted decision trees' werden die Klassifizierungen wesentlich stabiler als mit einzelnen Bäumen. Durch die Mittelung der Einzelentscheidungen in (9.45) ergibt sich auch ein Maß für die Wahrscheinlichkeit der richtigen Einordnung. Die Eigenschaften scheinen durchaus mit Neuronalen Netzen vergleichbar oder vielleicht sogar überlegen zu sein.

9.6 Stützvektormaschinen

Das Konzept einer so genannten "Stützvektormaschine (SVM)" ('support vector machine') greift die Idee auf, dass eigentlich nur Merkmalvektoren in der Nähe der Trennung zwischen den Klassen wesentlich sind: aus einem Trainingsdatensatz werden die Vektoren, die im wesentlichen die Trennung definieren, als "Stützvektoren" ausgewählt.

Im Folgenden werden wir zunächst die lineare Variante der SVM besprechen und dann die vielleicht interessantere Variante für die Anwendung auf nicht linear separierbare Klassen.

[2] http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

9.6.1 Lineare SVM-Klassifikation

Wir gehen zunächst von zwei disjunkten Klassen wie in Abb.9.31 aus. Die Klassen sollen durch eine lineare Diskriminante, also eine Hyperebene wie in Abschnitt 9.3 eingeführt, getrennt werden. Die Lage der diskriminierenden Hyperebene soll nun so optimiert werden, dass die nächsten Trainingsvektoren in beiden Klassen maximal von der Ebene entfernt sind, dass also der Trennungsstreifen möglichst breit wird. Ein solches Trainingsziel führt zu einer optimalen Generalisierungsfähigkeit.

Die Ränder des Trennungsstreifens sind zwei parallele Hyperebenen, die durch die Stützvektoren festgelegt werden sollen. Offensichtlich braucht man in m Dimensionen mindestens $m+1$ Stützvektoren. Zum Beispiel können m Vektoren eine Ebene festlegen und der verbleibende Vektor den Abstand der beiden Ebenen (siehe den zwei-dimensionalen Fall in Abb. 9.31). Die Aufgabe ist also, die maximale Trennung und die mindestens $m+1$ Stützvektoren zu bestimmen.

Die Diskriminante wird analog zu der Fisher-Diskriminante (Abschnitt 9.3.2) definiert:

$$\vec{w}^T \vec{x} + b = 0. \tag{9.46}$$

Wenn $\vec{w}$ ein Einheitsvektor ist, gibt b den Abstand vom Ursprung an, wenn im allgemeinen $\vec{w}$ kein Einheitsvektor ist, ist der Abstand von Ursprung durch $b/|\vec{w}|$ gegeben. Die beiden Randhyperebenen sollen in der Form

$$\vec{w}^T \vec{x} + b = \pm 1. \tag{9.47}$$

gegeben sein, was die Skala für $\vec{w}$ und b festgelegt. Dann ist der Abstand der Randebenen zur Diskriminante $d = 1/|\vec{w}|$. Für zwei Vektoren $\vec{x}^{(1)}$ und $\vec{x}^{(2)}$, die jeweils zu einer Randebene weisen, gilt:

$$\vec{w}^T(\vec{x}^{(1)} - \vec{x}^{(2)}) = 2. \tag{9.48}$$

Der tatsächliche Abstand zwischen den Hyperebenen ist

$$\frac{\vec{w}^T}{|\vec{w}|}(\vec{x}^{(1)} - \vec{x}^{(2)}) = \frac{2}{|\vec{w}|} = 2\,d. \tag{9.49}$$

Für alle Vektoren $\vec{x}$ gilt

$$|\vec{w}^T \vec{x} + b| \;\geq\; 1, \tag{9.50}$$

und zwar je nach Klassenzugehörigkeit

$$\vec{w}^T \vec{x} + b \;\geq\; +1 \quad \text{oder} \quad \vec{w}^T \vec{x} + b \;\leq\; -1. \tag{9.51}$$

Eine Testgröße für die Klassenzugehörigkeit wird deshalb durch folgende Funktion definiert:

$$y = y(\vec{x}) = \operatorname{sgn}\left(\vec{w}^T \vec{x} + b\right) = \pm 1 \tag{9.52}$$

Um einen möglichst großen Abstand der Randebenen zu bekommen, muss nach Gleichung (9.49) der Betrag des Normalenvektors minimiert werden,

$$|\vec{w}| = \text{Minimum}. \tag{9.53}$$

Dabei sollen gleichzeitig die Ereignisse beider Klassen außerhalb des Trennungsstreifens bleiben:

$$\text{Nebenbedingung}: \quad |\vec{w}^T \vec{x}_i + b| = y_i \left(\vec{w}^T \vec{x}_i + b\right) \geq 1, \quad i = 1, \ldots, N. \tag{9.54}$$

Die N Nebenbedingungen können mit der Methode der Lagrange-Multiplikatoren in eine 'Zielfunktion' einbezogen werden:

$$L(\vec{w}, b, \vec{\alpha} | \vec{x}_i,\, i = 1, \ldots, N) = \frac{1}{2}|\vec{w}|^2 - \sum_{i=1}^{N} \alpha_i \left(y_i \left(\vec{w}^T \vec{x}_i + b\right) - 1\right) \tag{9.55}$$

Diese Funktion soll bezüglich den Parametern $\vec{w}, b$ bei festem $\vec{\alpha}$ minimiert werden. Aus dem Verschwinden der Ableitungen,

$$\frac{\partial L}{\partial w_j} = 0 \quad (j = 1, \ldots, m); \qquad \frac{\partial L}{\partial b} = 0, \tag{9.56}$$

ergibt sich:

$$\vec{w} = \sum_{i=1}^{N} \alpha_i\, y_i\, \vec{x}_i \quad \text{und} \quad \sum_{i=1}^{N} \alpha_i\, y_i = 0. \tag{9.57}$$

Die Zwangbedingungen in (9.55) führen zu der Sattelpunkt-Bedingung (bezüglich der α_i), der Kuhn-Karush-Tucker-Bedingung:

$$\alpha_i \left\{y_i \left(\vec{w}^T \vec{x}_i + b\right) - 1\right\} = 0, \quad \forall\, i = 1, \ldots, N. \tag{9.58}$$

Das bedeutet, dass die α_i nur dann ungleich 0 sein können, wenn der Ausdruck in der geschweiften Klammer 0 ist, was aber nur für die Punkte auf dem Rand des Trennstreifens der Fall ist. Damit tragen nur die Merkmalsvektoren $\vec{x}_i$ mit $\alpha_i \neq 0$, die alle auf den Rändern liegen und Stützvektoren (support vectors) genannt werden, zu der Definition von $\vec{w}$ in (9.57) bei:

$$\vec{w} = \sum_{i=1}^{N_{SV}} \alpha_i\, y_i\, \vec{x}_i \tag{9.59}$$

Dabei geht die Summe nur über die N_{SV} Stützvektoren.

Der Ausdruck für den Normalenvektor $\vec{w}$ in (9.57) enthält die bisher noch nicht bestimmten Lagrange-Multiplikatoren α_i. Die Ausdrücke in (9.57) werden in die Formel für L in (9.55) eingesetzt, was nach einiger Rechnung ergibt:

$$L(\vec{w}, b, \vec{\alpha}\,) \;\rightarrow\; L_D(\vec{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j\, y_i y_j\, \vec{x}_i^T \vec{x}_j \tag{9.60}$$

mit den Nebenbedingungen:

$$\sum_{i=1}^{N} \alpha_i\, y_i = 0 \quad \text{und} \quad \alpha_i \geq 0. \tag{9.61}$$

Damit ist das Problem jetzt auf das sogenannte 'duale Problem', die Maximierung von $L_D(\vec{\alpha})$ bezüglich $\vec{\alpha}$ unter den Nebenbedingungen (9.61) zurückgeführt, das mit

numerischen Methoden gelöst werden kann. Mit der Lösung für die α_i ist der Normalenvektor der Trennebene vollkommen bestimmt:

$$\vec{w} = \sum_{i=1}^{N_{SV}} \alpha_i \, y_i \, \vec{x}_i \tag{9.62}$$

Nur die Stützvektoren auf dem Rand des Trennbereiches tragen zur Festlegung von $\vec{w}$ bei. Mit einem der Stützvektoren, zum Beispiel auf der '+'-Seite, kann jetzt noch b berechnet werden:

$$\vec{w}^T \, \vec{x}_{SV+} + b = +1 \quad \Rightarrow b = 1 - \vec{w}^T \, \vec{x}_{SV+} \tag{9.63}$$

Damit kann für jeden zu klassifizierenden Vektor $\vec{x}$ die Entscheidungsfunktion y in (9.52) bestimmt werden:

$$y = y(\vec{x}) = \operatorname{sgn}\left(\vec{w}^T \, \vec{x} + b\right) = \operatorname{sgn}\left(\sum_{i=1}^{N_{SV}} \alpha_i \, y_i \, \vec{x}_i^T \, \vec{x} + b\right) = \pm 1 \tag{9.64}$$

Bemerkenswert ist, dass nur Skalarprodukte des Testvektors mit den Stützvektoren zu berechnen und linear zu kombinieren sind. Die Tatsache, dass die Merkmalsvektoren nur in Skalarprodukten auftreten, macht man sich für eine Erweiterung des Merkmalsraumes in höhere Dimensionen mit einem verallgemeinerten Skalarprodukt zu Nutze, um auch nicht linear-separable Probleme zu lösen (siehe folgender Abschnitt).

Ohne hier in Details zu gehen, sei noch angemerkt, dass mit der linearen SVM auch moderat überlappende Klassen geteilt werden können, indem man die strikten Zwangsbedingungen (9.54) durch zusätzliche Terme mit so genannten 'Schlupfvariablen' aufweicht.

9.6.2 Nichtlineare Erweiterung mit Kernelfunktionen

Der oben beschriebene Algorithmus klassifiziert die Daten mit Hilfe einer linearen Funktion. Diese ist jedoch nur optimal, wenn auch das zu Grunde liegende Klassifikationsproblem linear separabel ist. In vielen Anwendungen ist dies aber nicht der Fall. Ein möglicher Ausweg ist, die Daten in einen Raum höherer Dimension abzubilden[3] (Abb.9.32):

$$\phi : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}, x \mapsto \phi(x) \qquad (d_1 < d_2). \tag{9.65}$$

Durch diese Abbildung wird die Anzahl möglicher linearer Trennungen erhöht (Theorem von Cover). Bei einer linearen Separierbarkeit gehen in die relevante Entscheidungsfunktion (9.64) die Datenpunkte $\vec{x}_i$ nur in Skalarprodukten ein. Daher ist es möglich, das Skalarprodukt $\vec{x}_i^T \vec{x}_j$ im Eingaberaum $\mathbb{R}^{d_1}$ durch ein Skalarprodukt $\langle \phi(x_i), \phi(x_j) \rangle$ im $\mathbb{R}^{d_2}$ zu ersetzen und stattdessen direkt zu berechnen. Die Kosten dieser Berechnung lassen sich sehr stark reduzieren, wenn eine positiv definite Kernel-Funktion als Skalarprodukt benutzt wird ('Kernel-Trick'):

$$k(\vec{x}_i, \vec{x}_j) = \langle \phi(\vec{x}_i), \phi(\vec{x}_j) \rangle \tag{9.66}$$

[3]Siehe auch http://de.wikipedia.org/wiki/Support_Vector_Machine

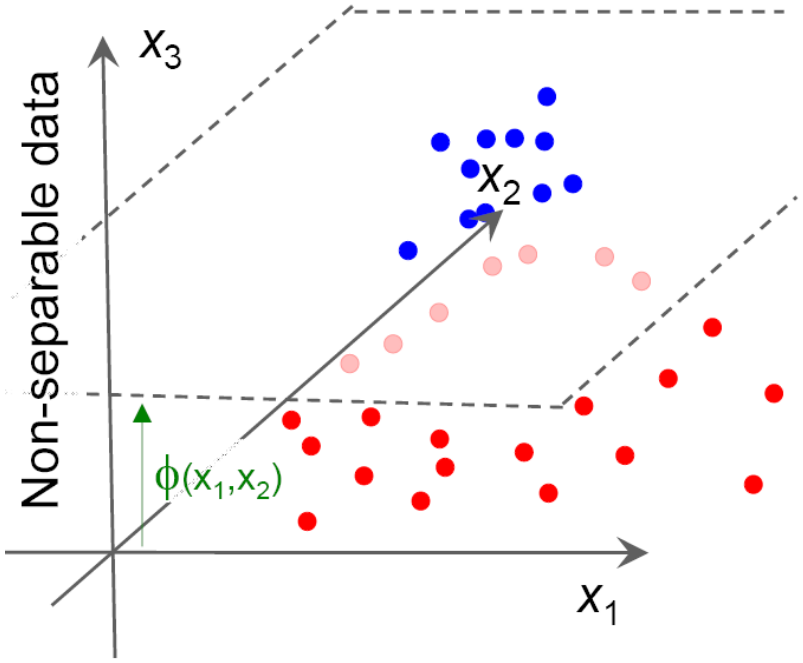

Abbildung 9.32: Beispiel eines in zwei Dimensionen nicht linear-separablen Datensatzes. Durch Transformationin eine höher dimensionalen Raum ist eine lineare Separation erreichbar.

Durch dieses Verfahren kann eine Hyperebene in einem höher-dimensionalen Raum implizit berechnet werden. Der resultierende Klassifikator hat die Form

$$y(\vec{x}) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i \, y_i \, k(\vec{x}_i, \vec{x}) + b\right). \tag{9.67}$$

Obwohl durch die Abbildung ϕ implizit ein möglicherweise unendlich-dimensionaler Raum benutzt wird, generalisieren SVM immer noch sehr gut.

Die Kern-Funktionen müssen symmetrisch und positiv definit sein. Beispiele sind:

- Polynomial (homogen): $k(\vec{x}, \vec{x}') = (\vec{x} \cdot \vec{x'})^d$
- Polynomial (inhomogen): $k(\vec{x}, \vec{x}') = (\vec{x} \cdot \vec{x'} + 1)^d$
- Radiale Basisfunktion: $k(\vec{x}, \vec{x}') = \exp\left(-\frac{|\vec{x}-\vec{x'}|^2}{2\sigma^2}\right)$
- Sigmoid-Funktion: $k(\vec{x}, \vec{x}') = \tanh(\kappa \vec{x} \cdot \vec{x'} + c)$, für $\kappa > 0$ und $c < 0$.

Beispiel: Mit einem einfachen Beispiel soll die Beziehung der Kernel-Funktionen zu Skalarprodukten in höher-dimensionalen Räumen erläutert werden: Es seien zwei Vektoren $\vec{x}_1$ und $\vec{x}_2$ in einem zwei-dimensionalen Merkmalsraum gegeben:

$$\vec{x}_1 = (x_{11}, x_{12}), \qquad \vec{x}_2 = (x_{21}, x_{22}) \tag{9.68}$$

Als Kern-Funktion wählen wir die inhomogene Polynomial-Funktion mit $d = 2$ aus:

$$\begin{aligned} k(\vec{x}_1, \vec{x}_2) &= (\vec{x}_1 \cdot \vec{x}_2 + 1)^2 \\ &= (x_{11}x_{21} + x_{12}x_{22} + 1)^2 \\ &= 2x_{11}x_{21} + 2x_{12}x_{22} + (x_{11}x_{21})^2 + (x_{12}x_{22})^2 + 2x_{11}x_{21}x_{12}x_{22} + 1 \end{aligned} \tag{9.69}$$

Die Zuordnung

$$\phi(\vec{x}_1) = \phi((x_{11}, x_{12})) = (1, \sqrt{2}x_{11}, \sqrt{2}x_{12}, x_{11}^2, x_{12}^2, \sqrt{2}x_{11}) \tag{9.70}$$

ist eine nicht-lineare Abbildung des 2-dimensionalen Raumes auf einen 6-dimensionalen Raum, in dem das Skalarprodukt durch die Kernel-Funktion definiert ist:

$$\langle \phi(\vec{x}_i), \phi(\vec{x}_j) \rangle = k(\vec{x}_i, \vec{x}_j) \tag{9.71}$$

Tatsächlich braucht die Transformation in die höhere Dimension (die auch unendlich sein kann, zum Beispiel bei der Gauss-Funktion) nicht durchgeführt zu werden, da man nur die Skalarprodukte berechnen muss, die durch die Kernel-Funktion gegeben sind.