Nichtlineare Klassifikatoren

Mustererkennung und Klassifikation, Vorlesung No. 11[1]

M. O. Franz

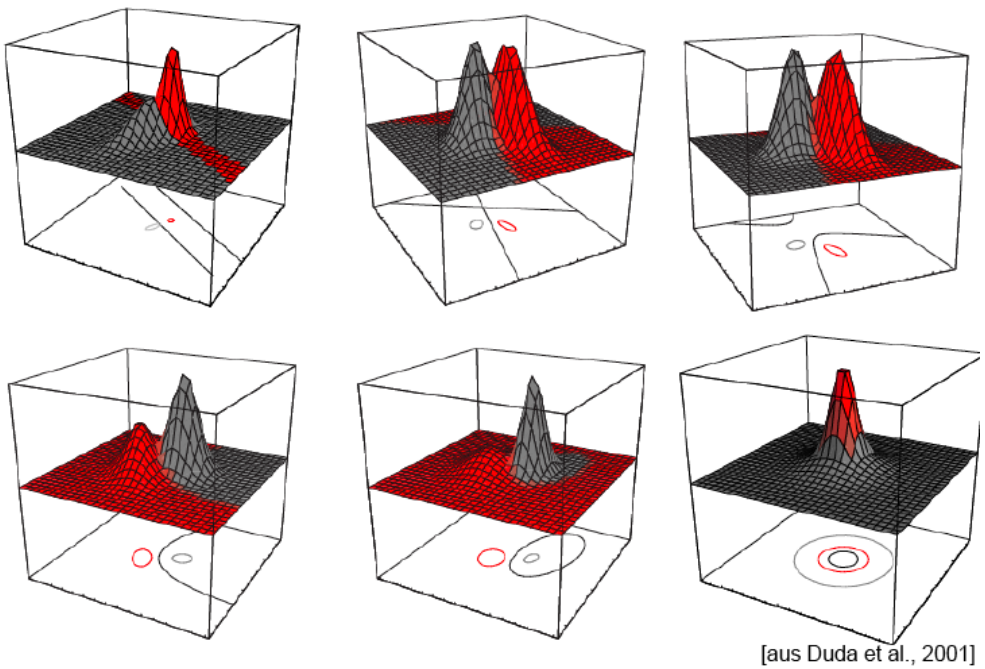
12.01.2008

[1] falls nicht anders vermerkt, sind die Abbildungen entnommen aus Duda et al., 2001.

Übersicht

Übersicht

Nichtlineare Probleme



[aus Duda et al., 2001]

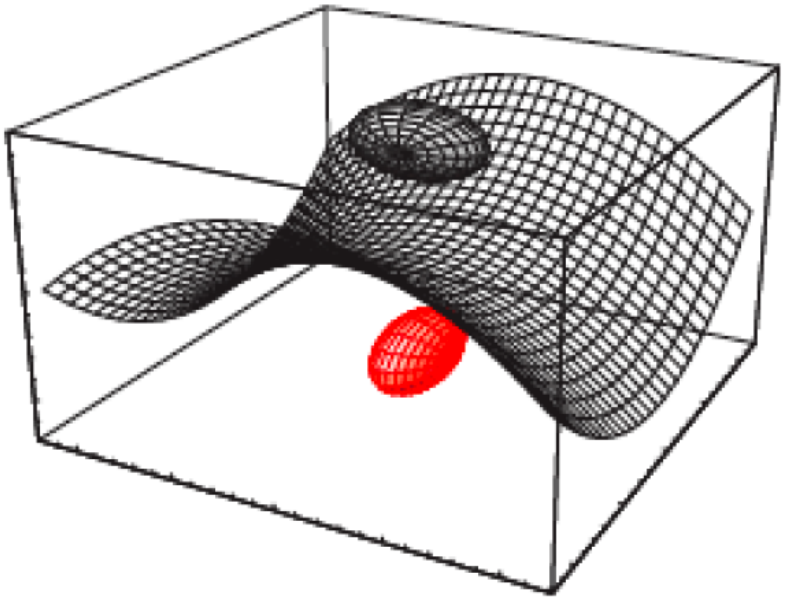
- Probleme mit nicht zusammenhängenden Entscheidungsregionen

Die bisher betrachteten Probleme waren entweder linear trennbar oder wurden ”zwangsweise” linear getrennt.

Diese Lösungen sind selten ideal. Problematische Fälle:

- nichtlinear trennbare Probleme
- Probleme, bei denen sich die Entscheidungsregionen geometrisch stark von Halbräumen unterscheiden

Nichtlineare Diskriminantenfunktionen



Die bisher betrachteten Trennebenen werden über lineare Diskriminanten-funktionen beschrieben
$(g(x) > 0? \, \omega_1 : \omega_2)$:

$$g(x) = w_0 + \sum_{i=1}^{d} w_i x_i = a^\top y$$

mit $a = (w_0, w_1, \ldots, w_d)^\top$ und
$y = (1, x_1, \ldots, x_d)^\top$.

Quadriken als Trennflächen sind quadratische Diskriminanten-funktionen

$$g(x) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d} \sum_{j=1}^{d} w_{ij} x_i x_j$$

beschreiben Kugeln, Ellipsoide und Hyperboloide.

Polynomiale Trennflächen

Komplexere Trennflächen können über Terme höherer Ordnung beschrieben werden (Polynomiale Trennflächen):

$$g(x) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d}\sum_{j=1}^{d} w_{ij} x_i x_j + \sum_{i=1}^{d}\sum_{j=1}^{d}\sum_{k=1}^{d} w_{ijk} x_i x_j x_k + \ldots$$

sind gewichtete Summen von Monomen $m(x)$:

- 1 Monom der Ordnung 0: 1
- d Monome der Ordnung 1: $x_1, x_2, \ldots, x_d$
- d^2 Monome der Ordnung 2: $x_1x_2, x_1^2, x_2^2, x_1x_3, \ldots$
- d^3 Monome der Ordnung 3: $x_1^3, x_1^2x_2, x_1x_2^2, x_1x_2x_3, \ldots$
- :
- d^n Monome der Ordnung n

Schreibweise als erweiterter Merkmalsvektor

Analog zum linearen Fall werden die Monome und ihre Gewichte in einen erweiterten Merkmals- und Gewichtsvektor geschrieben:

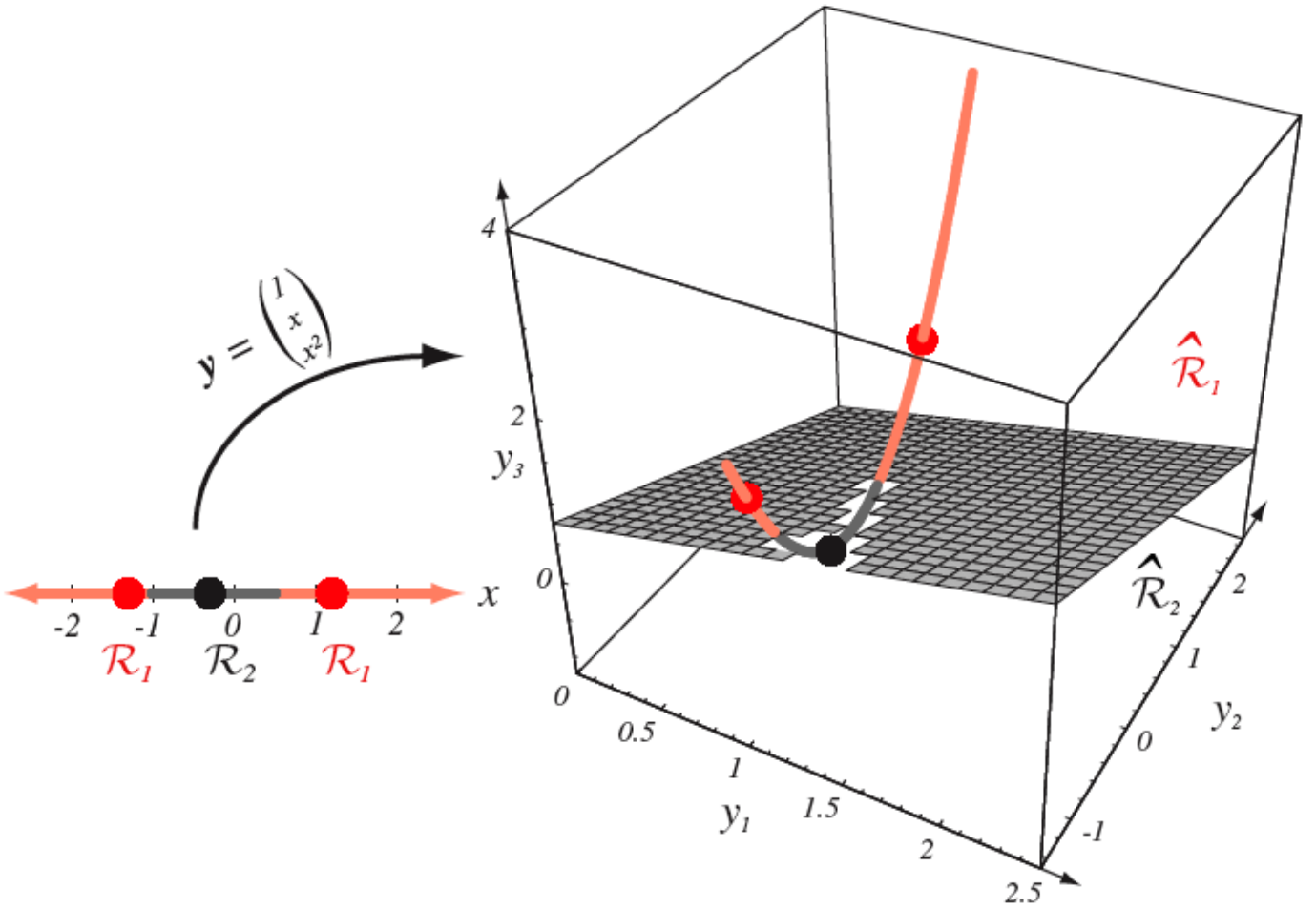
$$y = \Phi(x) = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_1^2 \\ x_1 x_2 \\ \vdots \\ x_1^3 \\ \vdots \end{bmatrix} \quad \text{und} \quad a = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_{11} \\ w_{12} \\ \vdots \\ w_{111} \\ \vdots \end{bmatrix} \quad \Rightarrow \quad g(x) = a^\top y = a^\top \Phi(x)$$

d.h. die Berechnung des erweiterten Merkmalvektors kann als eine nichtlineare Abbildung $\Phi(x)$ aus dem d-dimensionalen Inputraum in einen $1 + d + d^2 + \cdots + d^n$-dimensionalen Raum (Merkmalsraum) betrachtet werden!

Beispiel (1)

Eindimensionaler Input x, quadratische Diskriminantenfunktion

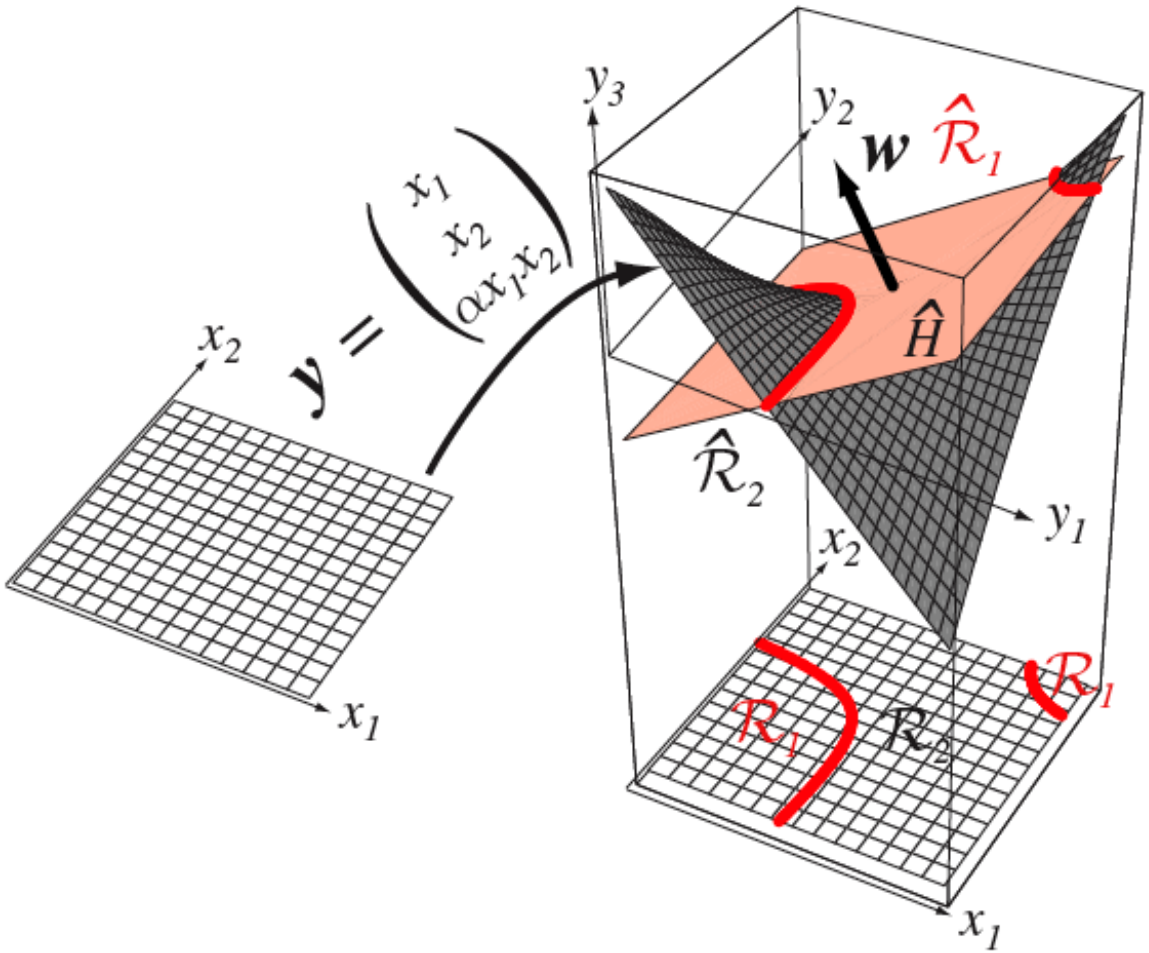
$$g(x) = a_1 + a_2 x + a_3 x^2$$



Beispiel (2)

Zweidimensionaler Input $x = (x_1, x_2)$, quadratische Diskriminantenfunktion

$$g(x) = a_1 x_1 + a_2 x_2 + a_3 x_1 x_2$$



Grundidee der Klassifikation mit nichtlinearen Basisfunktionen

1. Beschreibe die Trennfläche als gewichtete Summe von m geeigneten nichtlinearen Basisfunktionen $\varphi_i(x)$ (z.B. Monome):

$$g(x) = \sum_{i=1}^{m} a_i \varphi_i(x).$$

   Oft wird m um einiges höher als d gewählt.

2. Betrachte die Berechnung des erweiterten Merkmalsvektors $y = (\varphi_1(x), \varphi_2(x), \ldots, \varphi_m(x))^\top$ als nichtlineare Abbildung $\Phi(x)$ aus dem d- dimensionalen Inputraum $\mathbb{R}^d$ in den m-dimensionalen Merkmalsraum $\mathbb{R}^m$.

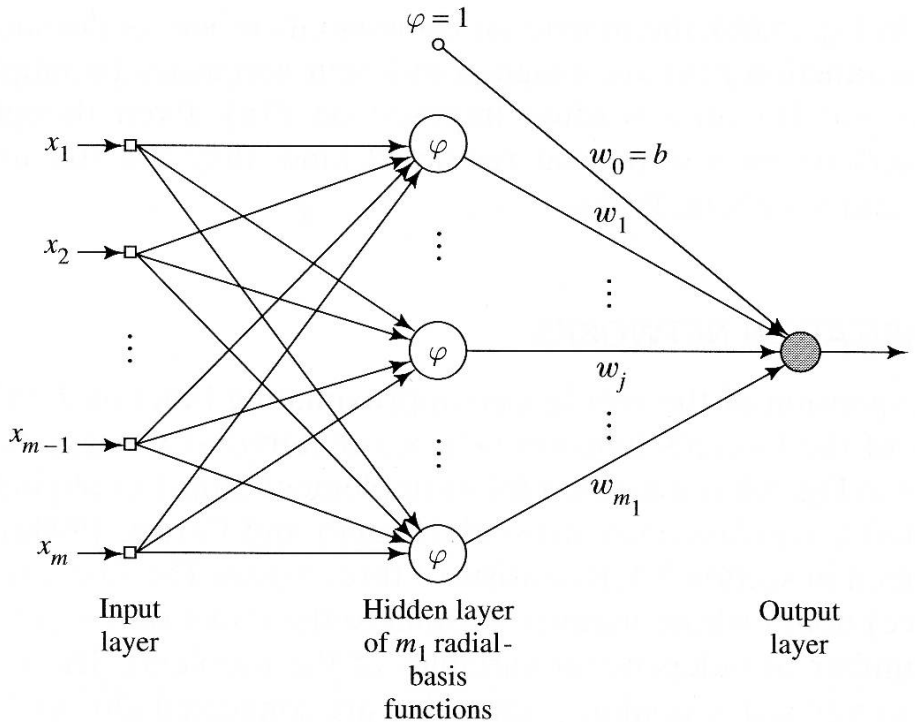
3. Nach Abbildung des Inputs in $\mathbb{R}^m$ führe lineare Klassifikation in $\mathbb{R}^m$ durch (z.B. Perzeptron oder kleinste Quadrate).

Klassifikation mit nichtlinearen Basisfunktionen - Eigenschaften

- Je mehr Basisfunktionen, desto komplexere Trennflächen können approximiert werden.
- Je mehr Basisfunktionen, desto wahrscheinlicher ist das Problem im Merkmalsraum $\mathbb{R}^m$ linear trennbar (*Covers Theorem zur Trennbarkeit von Mustern*, s. Haykin 99).
- Je mehr Basisfunktionen, desto mehr Trainingsdaten werden i.A. zur sicheren Bestimmung der Trennfläche gebraucht (z.B. ist die Zahl der Monome 3. Ordnung bei 10-dimensionalem Input $10^3 = 1000$, also braucht man ein Mehrfaches von 2000 an Trainingsdatenpunkten).
- Je mehr Basisfunktionen, desto mehr kann sich die Trennfläche an die Trainingsdaten anpassen, desto höher ist die Gefahr der Überanpassung (Overfitting) und damit schlechterer Leistung bei bisher ungesehenen neuen Datenpunkten (Generalisierung).

Übersicht

Netzwerke aus radialen Basisfunktionen (RBF-Netze)



Spezielle Wahl der Basisfunktionen als radiale Basisfunktionen (RBF)

$$\varphi_i(x) = \varphi_i(\|x - x_i\|),$$

d.h. jede Basisfunktion φ_i hängt nur vom Abstand von einem einzigen Trainingsdatenpunkt x_i ab.

Es gibt immer gleich viele Basisfunktionen wie Trainingsdaten.

Gewichtete Summen von RBFs heißen RBF-Netzwerke:

$$f(x) = \sum_{i=1}^{n} w_i \varphi_i(\|x - x_i\|)$$

Beispiele für RBFs

Der Abstand eines Inputdatenpunktes x zum Trainingsdatenpunkt x_i sei $r_i = \|x - x_i\|$

- Gaußfunktion:

$$\varphi_i(x) = e^{-\frac{r_i^2}{2\sigma^2}} \quad \text{mit Standardabweichung} \quad \sigma > 0$$

- Inverse Multiquadriken:

$$\varphi_i(x) = \frac{1}{\sqrt{r_i^2 + c^2}} \quad \text{mit} \quad c > 0$$

- Multiquadriken:

$$\varphi_i(x) = \sqrt{r_i^2 + c^2} \quad \text{mit} \quad c > 0$$

Klassifikation mit RBF-Netzen und der Methode der kleinsten Quadrate

RBF-Netze werden meist zur Schätzung von kontinuierlichen skalaren Outputwerten (Regression) verwendet. Sie können aber analog zur Methode der kleinsten Quadrate bei linearen Klassifikatoren auch zur Klassifikation eingesetzt werden:

1. Die Diskriminantenfunktion $g(x)$ bzw. die Trennfläche wird als RBF-Netz dargestellt:

$$g(x) = \sum_{i=1}^{n} w_i \varphi_i(\|x - x_i\|)$$

2. Die RBFs werden als nichtlineare Transformation in den n-dimensionalen Merkmalsraum $\mathbb{R}^n$ (n: Anzahl der Trainingsdatenpunkte) betrachtet.
3. Im $\mathbb{R}^n$ wird eine lineare Trennfläche mit der Methode der kleinsten Quadrate gefunden.

Beispiel: Klassifikation mit Gauß-RBFs (1)

Schritt 1: Beschreibe die Trennfläche als gewichtete Summe von m geeigneten nichtlinearen Basisfunktionen $\varphi_i(x)$.

Für unser Beispiel wählen wir für jeden der n Trainingsdatenpunkte eine Gauß-RBFs mit $\sigma = 1$ als Basisfunktion:

$$\varphi_i(x) = e^{-\frac{\|x-x_i\|^2}{2}}$$

Damit wird die Diskriminantenfunktion beschrieben als

$$g(x) = \sum_{i=1}^{n} w_i e^{-\frac{\|x-x_i\|^2}{2}}$$

Gesucht ist ein Gewichtssatz von w_i, der die Trainingsbeispiele korrekt klassifiziert.

Beispiel: Klassifikation mit Gauß-RBFs (2)

Schritt 2: Berechnung des erweiterten Merkmalsvektors im $\mathbb{R}^n$

$$y = \Phi(x) = \begin{bmatrix} e^{-\frac{\|x-x_1\|^2}{2}} \\ e^{-\frac{\|x-x_2\|^2}{2}} \\ \vdots \\ e^{-\frac{\|x-x_n\|^2}{2}} \end{bmatrix} \in \mathbb{R}^n \quad \Rightarrow \quad g(x) = a^\top y = a^\top \Phi(x)$$

Als nächstes müssen wir die Daten aus Klasse ω_2 am Ursprung in $\mathbb{R}^n$ spiegeln, d.h. $\Phi(x) = -\Phi(x)$ für alle $x_i \in \omega_2$.

Weiterhin muß für jeden Datenpunkt x_i ein skalierter Abstand (margin) b_i in $\mathbb{R}^n$ festgelegt werden, z.B. einheitlich $b_i = 1$ für alle x_i.

Beispiel: Klassifikation mit Gauß-RBFs (3)

Schritt 3: Anwendung der Methode der kleinsten Quadrate in $\mathbb{R}^n$

Aufgabe: Finde einen Gewichtsvektor a in $\mathbb{R}^n$, so daß die zugehörige Entscheidungsebene zu jedem nichtlinear abgebildeten Datenpunkt $\Phi(x_i)$ den skalierten Abstand b_i hat, d.h. $\Phi(x_i)^\top a = b_i$.

Matrixschreibweise:
$$\begin{pmatrix} \Phi(x_1)^\top \\ \Phi(x_2)^\top \\ \vdots \\ \Phi(x_n)^\top \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix} \quad \text{bzw.} \quad Ya = b$$

bzw.
$$\begin{pmatrix} 1 & e^{-\frac{\|x_2 - x_1\|^2}{2}} & \dots & e^{-\frac{\|x_n - x_1\|^2}{2}} \\ e^{-\frac{\|x_1 - x_2\|^2}{2}} & 1 & \dots & e^{-\frac{\|x_n - x_2\|^2}{2}} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-\frac{\|x_1 - x_n\|^2}{2}} & e^{-\frac{\|x_2 - x_n\|^2}{2}} & \dots & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_d \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix}$$

Beispiel: Klassifikation mit Gauß-RBFs (4)

Im Gegensatz zum linearen Fall ist

- die Matrix Y quadratisch, da es genauso viele Basisfunktionen wie Datenpunkte gibt. Damit existiert eine inverse Matrix Y^{-1}, falls Y invertierbar ist.
- Y bei Gauß-RBFs (zumindest theoretisch) immer invertierbar, falls alle Trainingsdatenpunkte unterschiedlich sind (s. Haykin 1999).

Unter diesen Bedingungen sind Inverse und Pseudoinverse gleich. Damit ist die Lösung der Aufgabe gegeben durch

$$a = Y^{\dagger} b = Y^{-1} b.$$

Für andere RBF-Typen oder bei numerischen Problemen (z.B. extrem nahe zusammenliegende x_i) muß u.U. immer noch die Pseudoinverse verwendet werden.

Übersicht

Probleme bei durch kleinste Quadrate gefundenen RBF-Netzen

- Es gibt exakt gleich viele Trainingsdatenpunkte wie Funktionen. Damit ist die Trennfläche in vielen Fällen nicht ausreichend überbestimmt.
- Es gibt keinen Mechanismus für verrauschte Probleme, der den Einfluß des Rauschens auf die Form der Trennfläche kontrolliert. In solchen Fällen wird die Trennfläche unnötig komplex, da auch das Rauschen mitmodelliert wird. I.A. haben solche verrauschten Trennflächen schlechtere Generalisierungseigenschaften.
- Kleinste-Quadrate-Schätzungen haben zwar potentiell eine sehr hohe Genauigkeit (d.h. einen kleinen Bias), reagieren aber sehr empfindlich auf Rauschen (d.h. hohe Varianz), insbesondere auf Ausreißer in den Daten.
- Das Kleinste-Quadrate-Kriterium weist oft einigen wenigen fehlerhaften Datenpunkten ein zu hohes Gewicht zu, da der Fehler quadriert wird.

Regularisierung

Um die Variabilität der Trennfläche einzuschränken, müssen die Freiheitsgrade der möglichen Lösungen sinnvoll eingeschränkt werden. Diesen Vorgang nennt man Regularisierung. Er erfordert Vorwissen über die Art der vorkommenden Lösungen oder eine vorherige Festlegung bestimmter, gewünschter Eigenschaften der Lösungen.

In unserem Fall wollen wir Lösungen, die

- nicht jedes Detail der Trainingsdaten nachmodellieren, sondern nur ihre ”wesentliche” Struktur.
- nicht die rauschbedingten Variationen in den Daten nachmodellieren.
- nicht einzelnen Ausreißern ein übermäßig hohes Gewicht zuweisen.

Regularisierung durch Beschränkung des Gewichtsvektors

Ansatz: Statt alleine den quadratischen Fehler zu minimieren, minimieren wir gleichzeitig die Länge $\|a\|$ des Gewichtsvektores. Dies geschieht durch das Zufügen eines Bestrafungstermes $\lambda\|a\|^2$ zum quadratischen Fehler:

$$J_\lambda(a) = \|e\|^2 + \lambda\|a\|^2 = \|Ya - b\|^2 + \lambda\|a\|^2$$

Der Faktor $\lambda > 0$ kontrolliert den Einfluß des Bestrafungsterms.

Vorteile:

- Lösungen, die einzelnen Datenpunkten ein zu hohes Gewicht zuweisen, werden unterdrückt $\Rightarrow$ geringere Empfindlichkeit gegen Ausreißer.
- Die Gewichte werden auf möglichst viele Datenpunkte verteilt. Damit werden die Lösungen automatisch glatter $\Rightarrow$ geringere Rauschempfindlichkeit, unwesentliche Details werden nicht mitmodelliert.

Regularisierte Lösung

Am Minimum muß die Ableitung des regularisierten quadratischen Fehlers $J_\lambda(a) = \|e\|^2 + \lambda\|a\|^2 = \|Ya - b\|^2 + \lambda\|a\|^2$ nach a gleich 0 sein, d.h.

$$\nabla J_\lambda = 2Y^\top(Ya - b) + 2\lambda a = 0$$

Umstellung führt auf modifizierte Normalengleichung:

$$(Y^\top Y + \lambda I)\, a = Y^\top b.$$

Da $Y^\top Y + \lambda I$ immer quadratisch und für $\lambda > 0$ nie singulär ist, existiert eine Inverse und damit immer eine Lösung:

$$a = (Y^\top Y + \lambda I)^{-1} Y^\top b$$