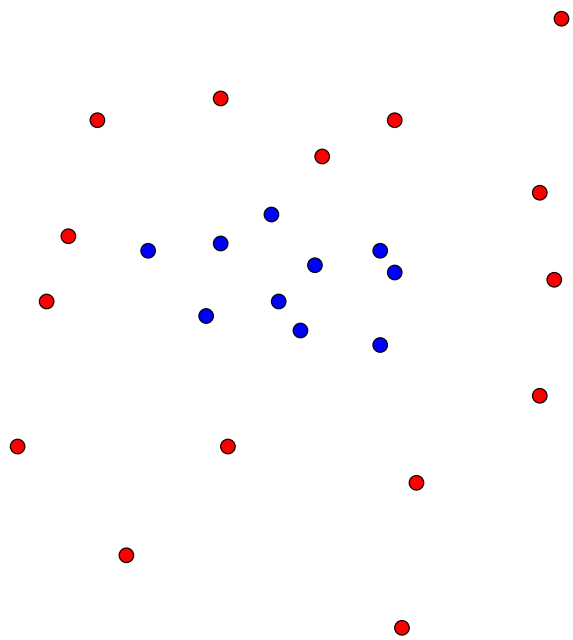
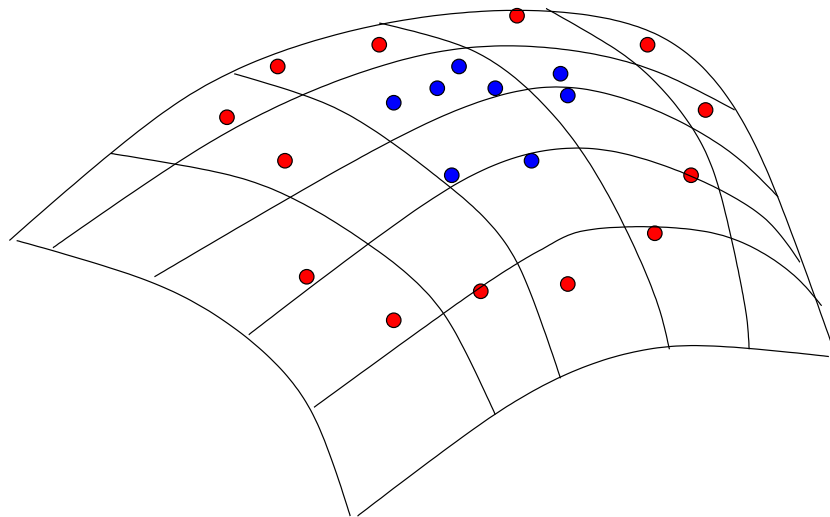
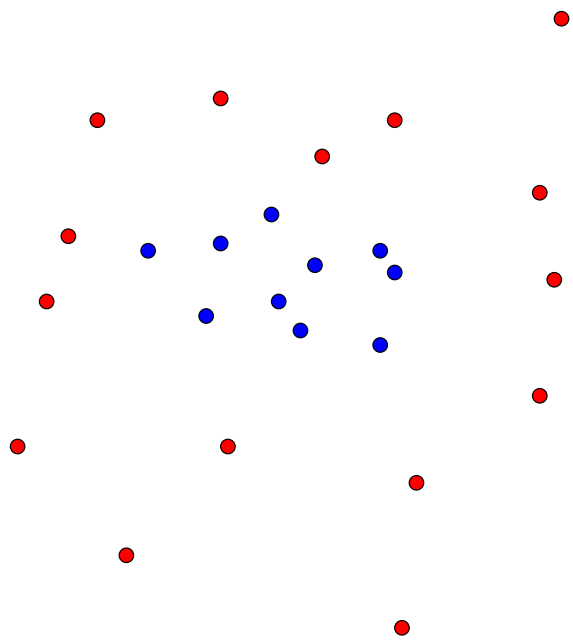
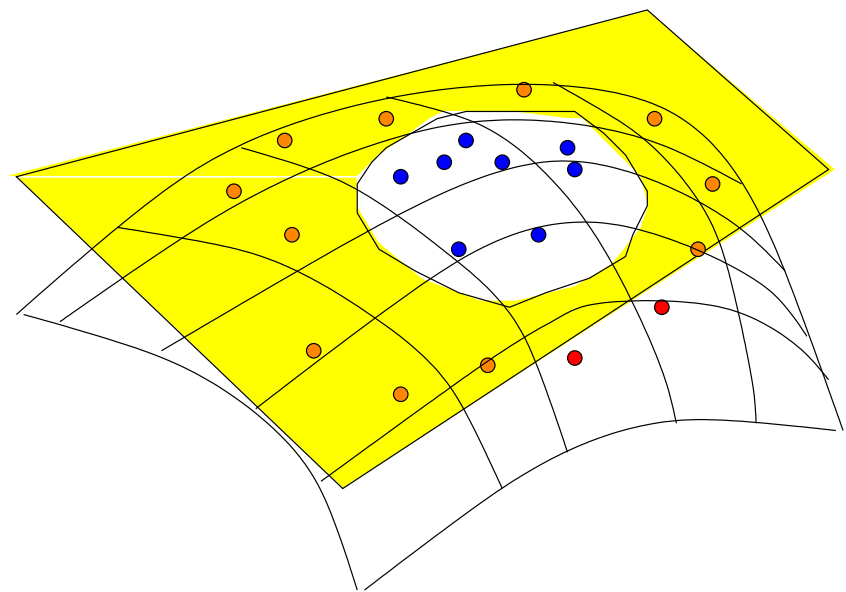
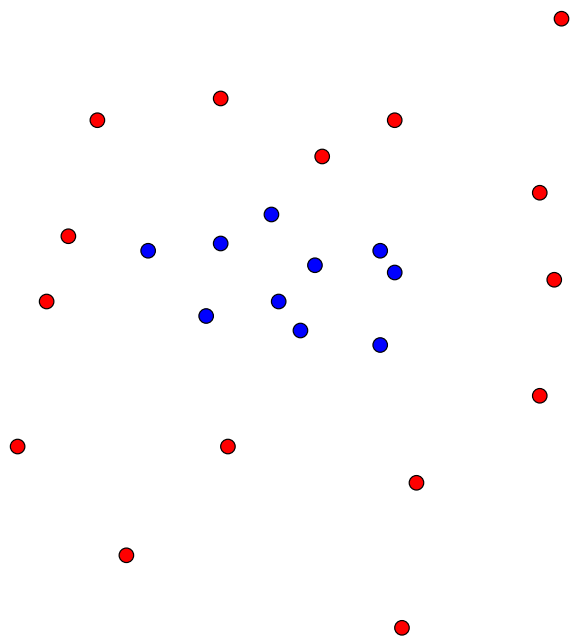


# **4 Support Vector Machines**







Bilde Datenpunkte ab in (möglicherweise hochdimensionalen) Vektorraum  $V$  (“**Feature-Raum**”)

$$\phi : x_j = (x_{j1}, \dots, x_{jn}) \mapsto (\phi_1(x_j), \dots, \phi_k(x_j), \dots)$$

Dort linear zu trennen gemäß Hard-Margin- oder Soft-Margin-Kriterium kann man durch Lösen des inversen Problems, d.h. maximiere

$$g(\xi) = \sum_{j=1}^m \xi_j - \frac{1}{2} \sum_{j,k=1}^m \xi_j \xi_k y_j y_k \langle \phi(x_j), \phi(x_k) \rangle$$

mit den entsprechenden Randbedingungen. Dazu müssen wir nicht wirklich in  $V$  rechnen, wir benötigen nur den sog. **Kernel**  $K : (x_j, x_k) \mapsto \langle \phi(x_j), \phi(x_k) \rangle$

Aus den KKT-Bedingungen erhalten wir u.a.  $\beta = \sum_{i=1}^m y_i \xi_i \phi(x_i) \in V$ .

Um ein neues  $x$  zu klassifizieren, berechnen wir das Vorzeichen von

$$\langle \phi(x), \beta \rangle + \beta_0$$

Das ist aber

$$\langle \phi(x), \sum_{i=1}^m y_i \xi_i \phi(x_i) \rangle + \beta_0 = \sum_{i=1}^m y_i \xi_i \langle \phi(x), \phi(x_i) \rangle + \beta_0 = \sum_{i=1}^m y_i \xi_i K(x, x_i) + \beta_0$$

Auch dazu braucht man also nur im  $\mathbb{R}^m$  und mit dem Kernel zu rechnen.

Die Dimension von  $\xi$  ist  $m$ , die von  $x$  und  $x_j$  ist  $n$ , die von  $V$  spielt keine Rolle, darf auch  $\infty$  sein.

## 4.1 Lernen mit Kernen

beliebte Kernels:

polynomial  $d$ -ten Grades:

$$K(x, z) = (1 + \langle x, z \rangle)^d$$

radiale Basis (Gauß-Kern):

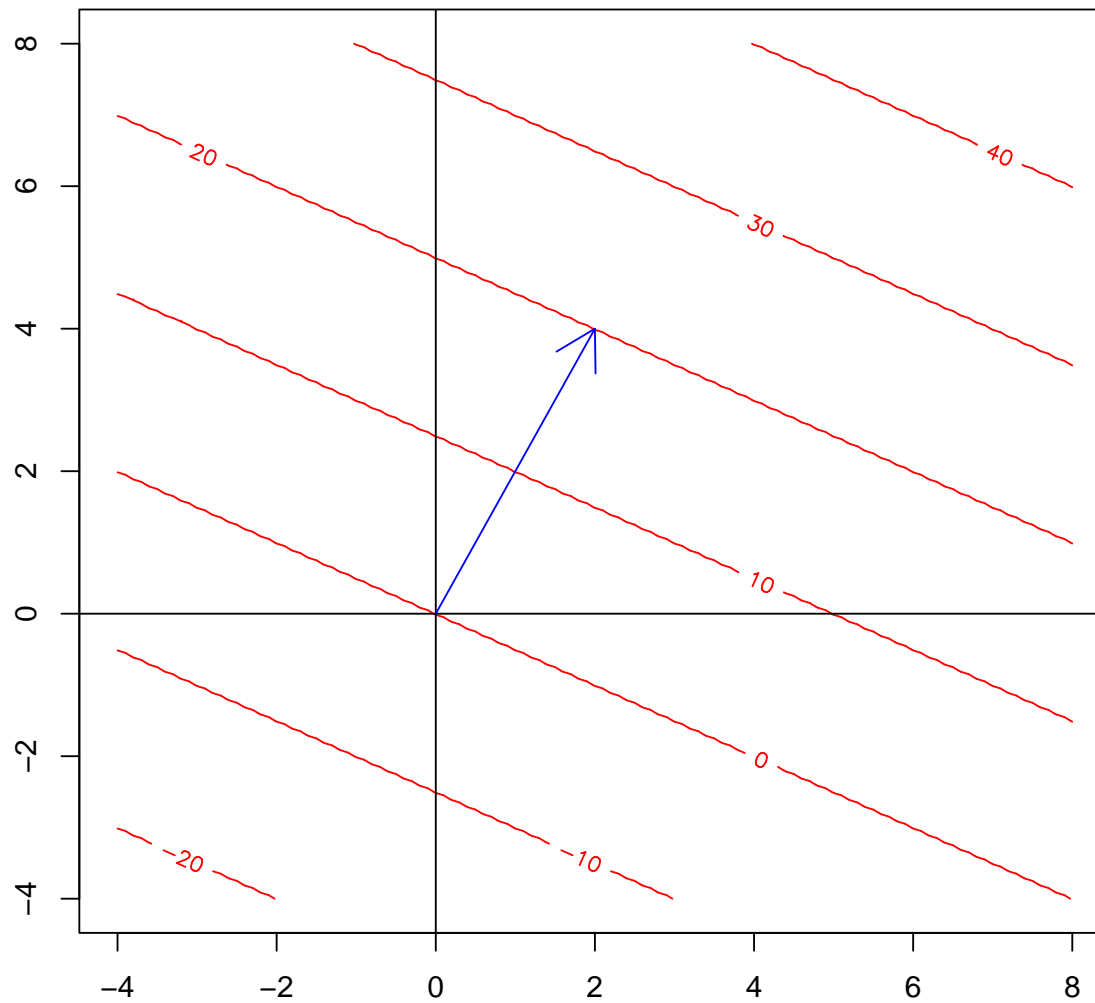
$$K(x, z) = \exp\left(-\frac{\|x - z\|^2}{c}\right)$$

sigmoid:

$$K(x, z) = \tanh(\kappa_1 \langle x, z \rangle + \kappa_2)$$

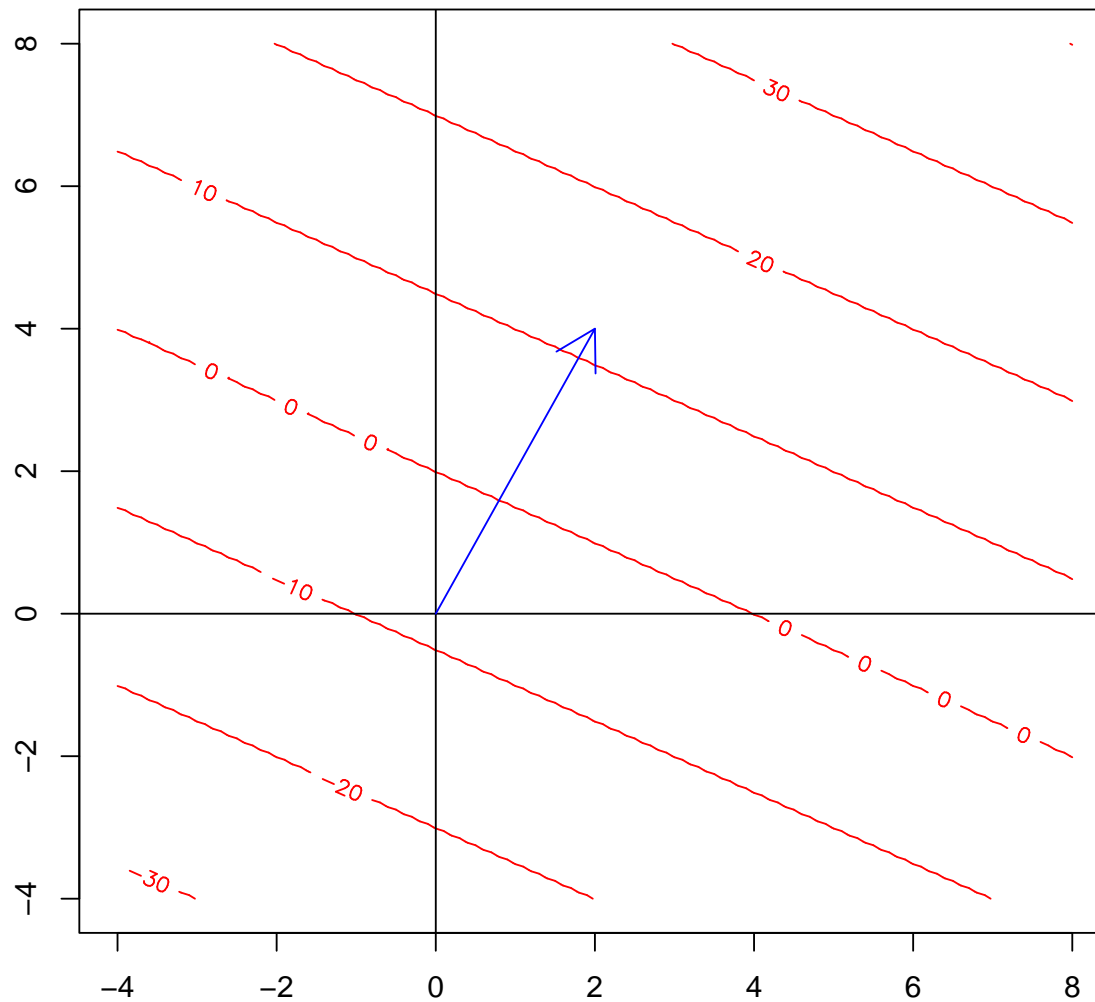
$$x = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$$

$$z \mapsto \langle x, z \rangle$$



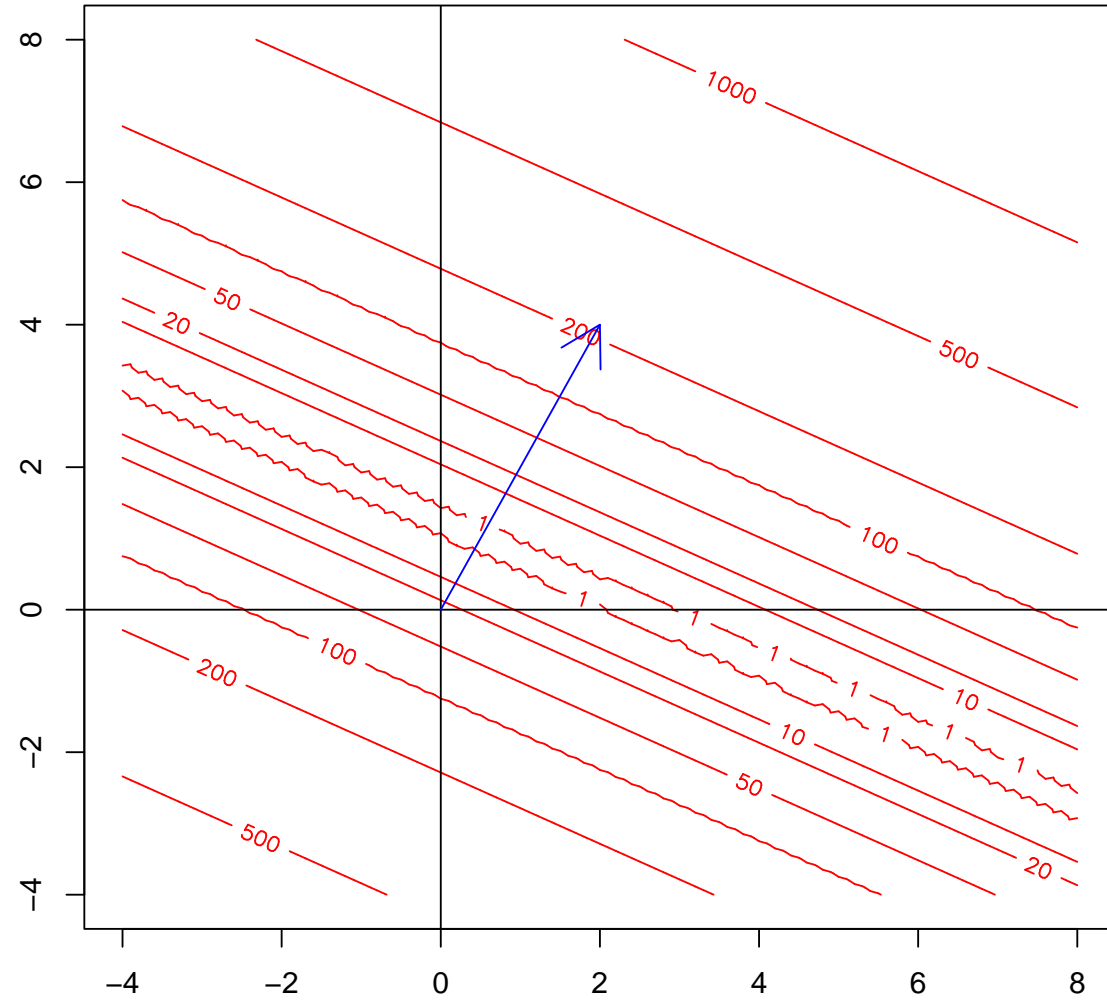
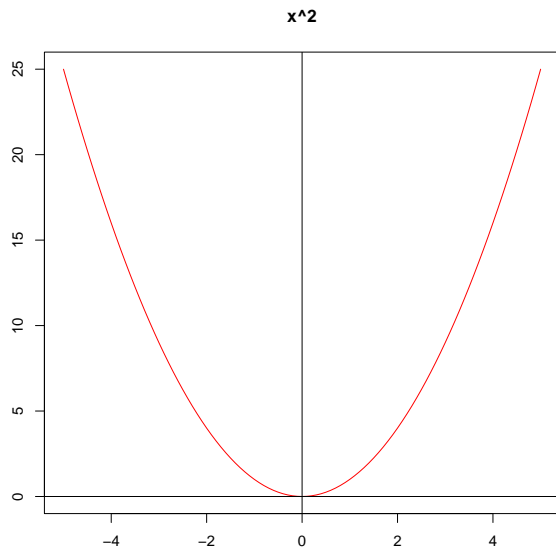
$$x = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$$

$$z \mapsto \langle x, z \rangle - 8$$



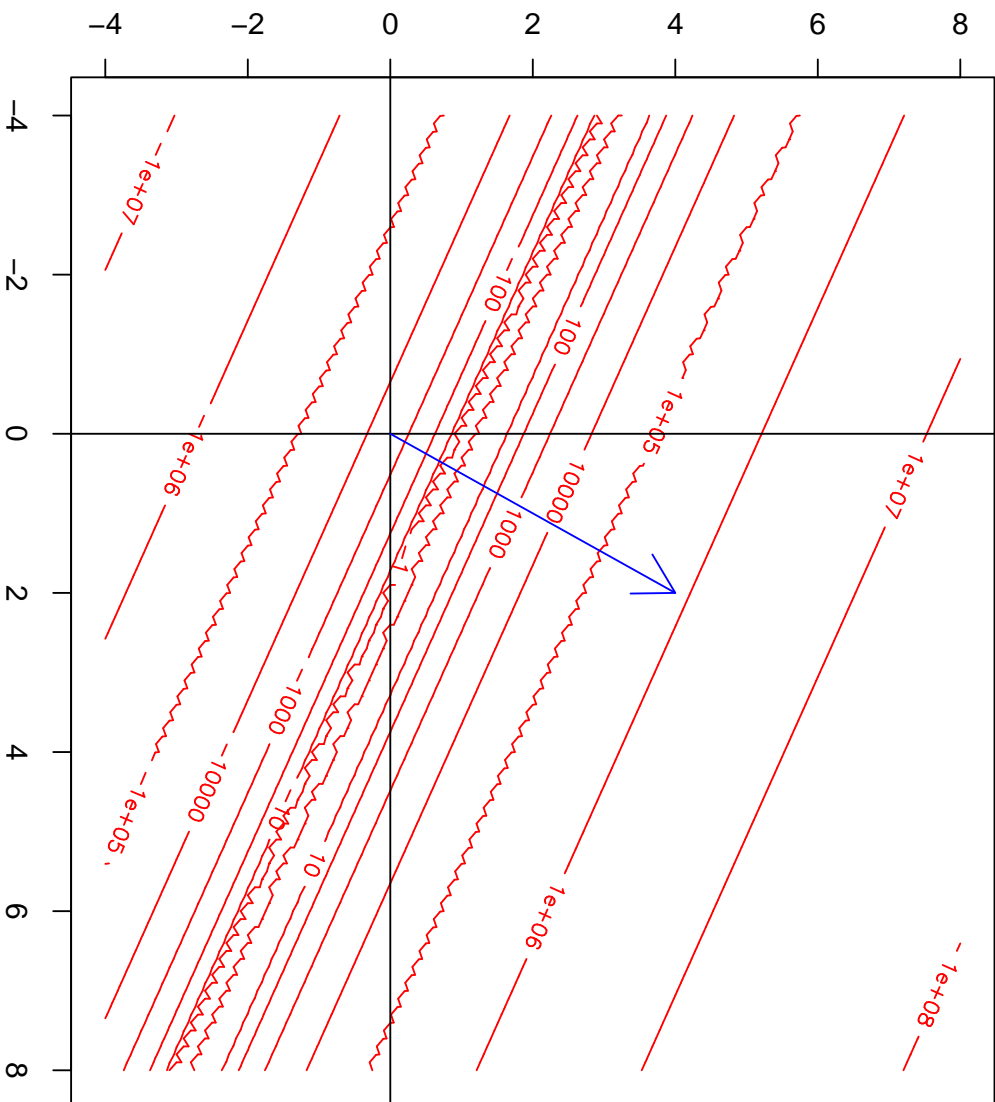
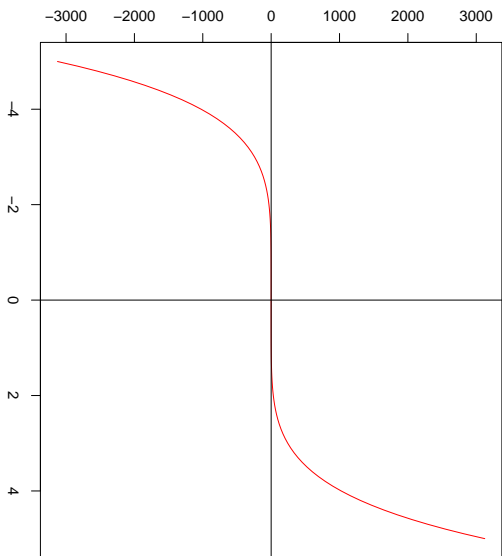
$$x = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$$

$$z \mapsto (\langle x, z \rangle - 5)^2$$



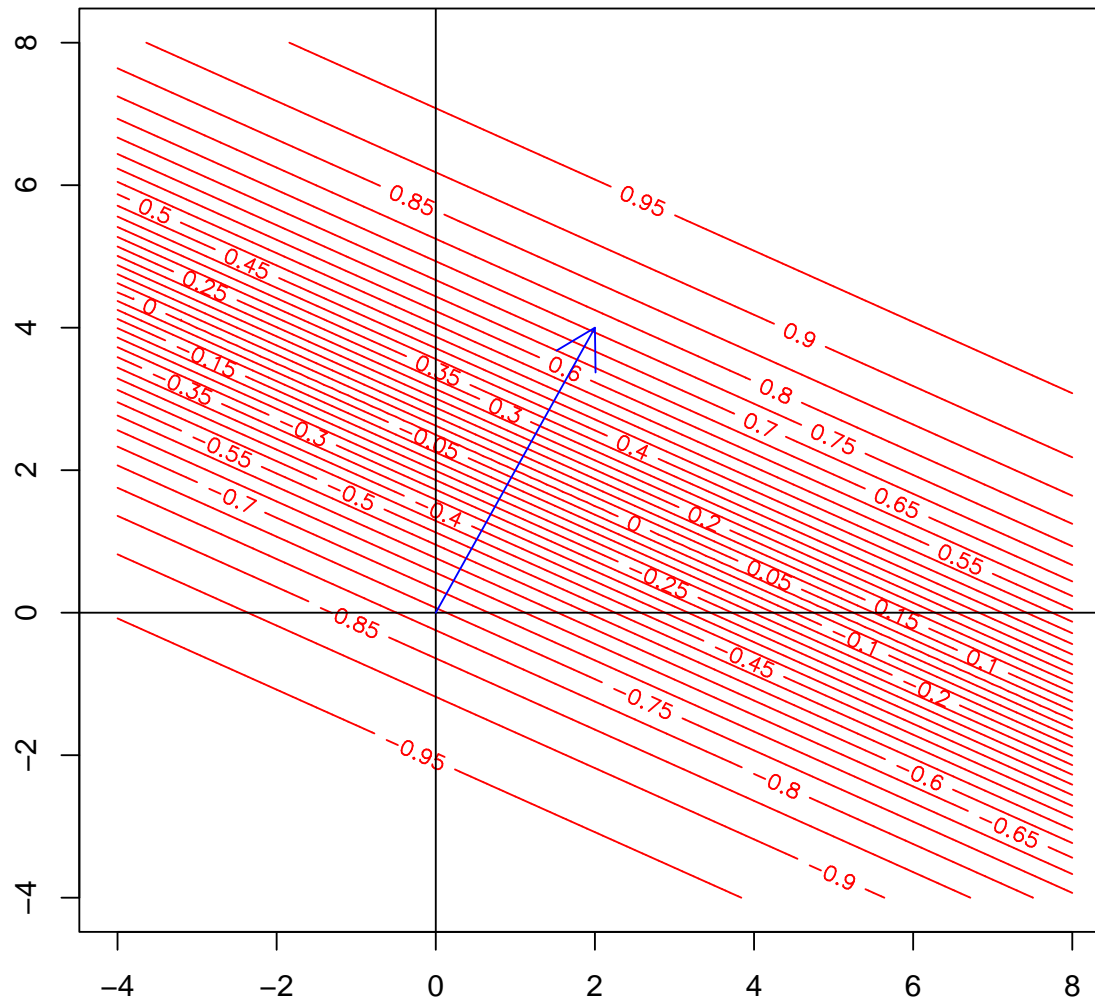
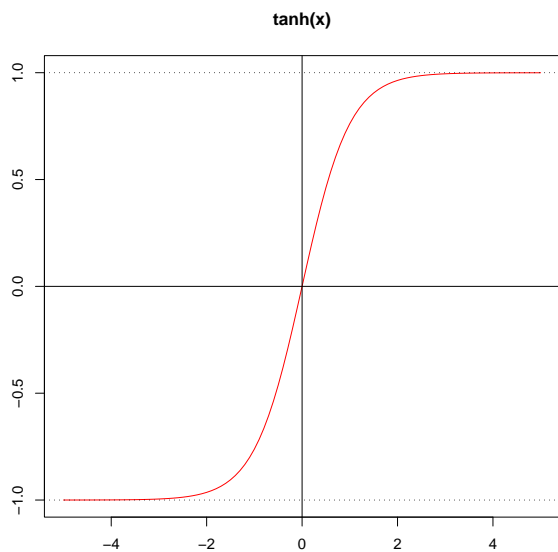
$$x = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$$

$$z \mapsto (\langle x, z \rangle - 5)^5$$



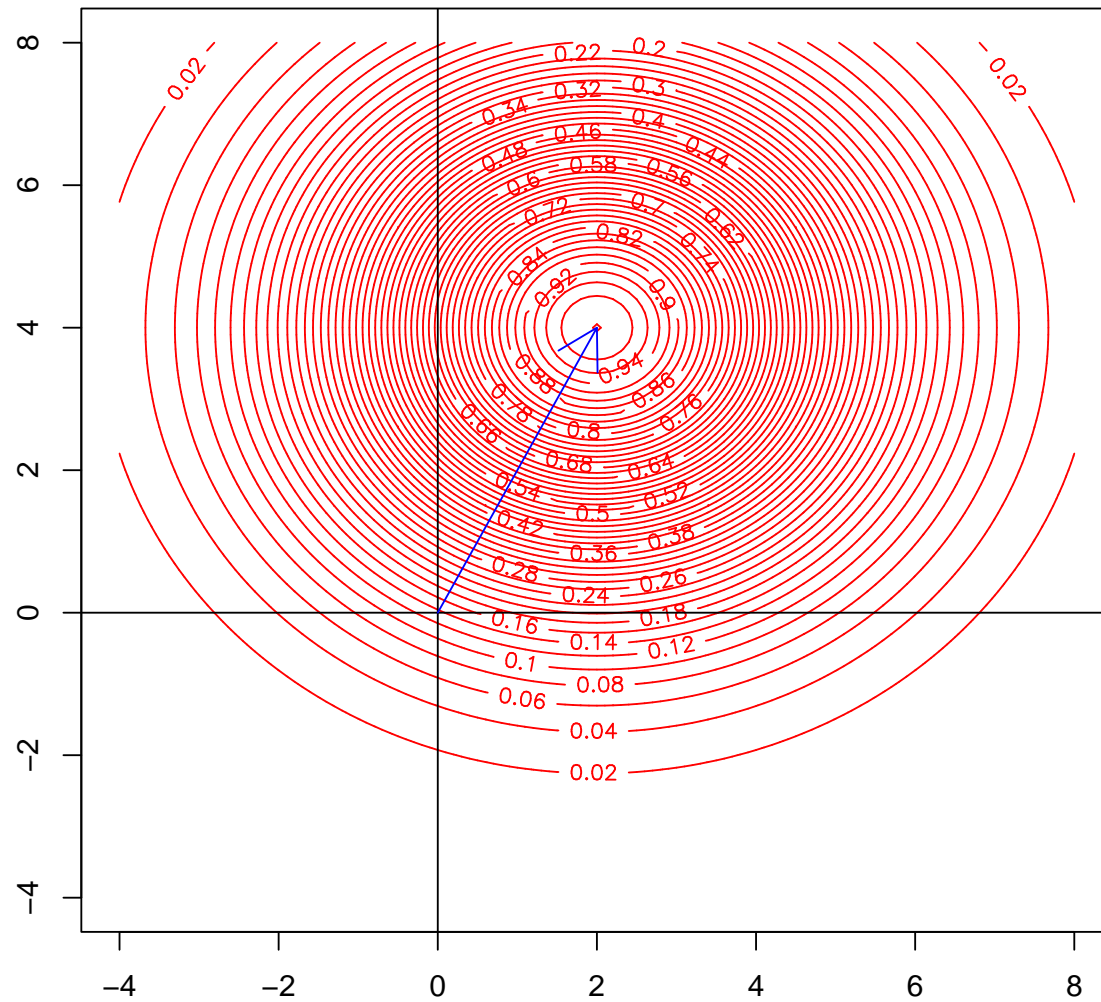
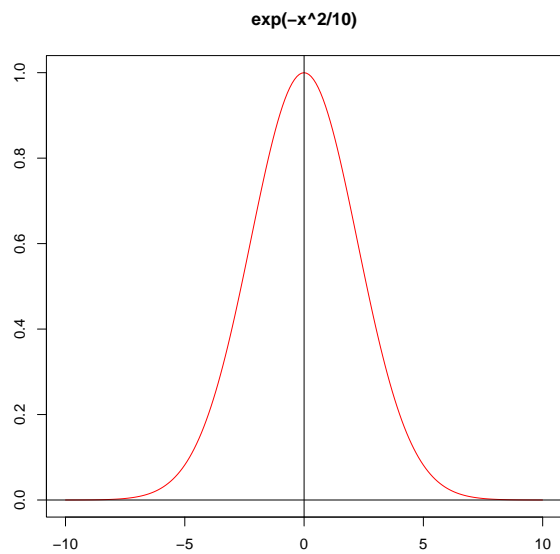
$$x = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$$

$$z \mapsto \tanh\left(\frac{\langle x, z \rangle}{10} - 1\right)$$

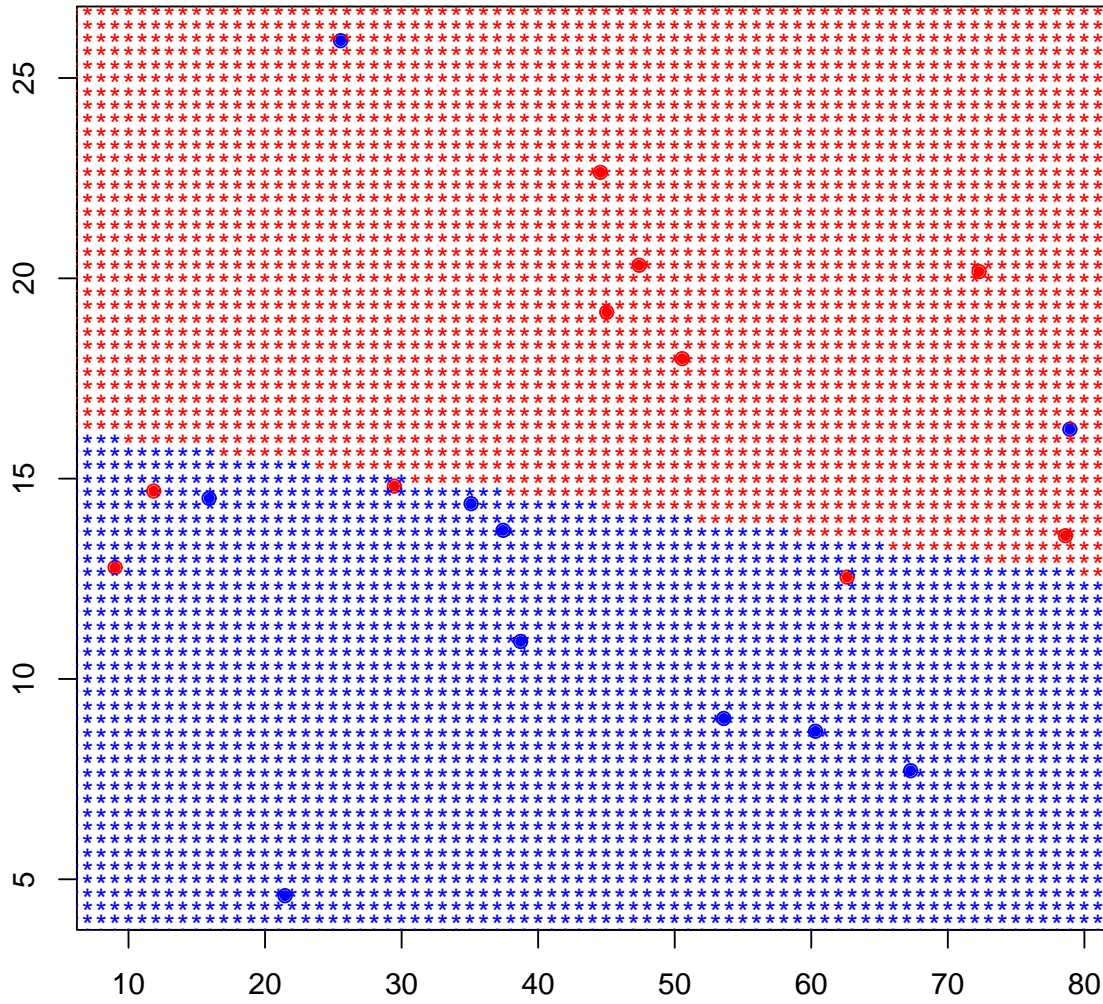


$$x = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$$

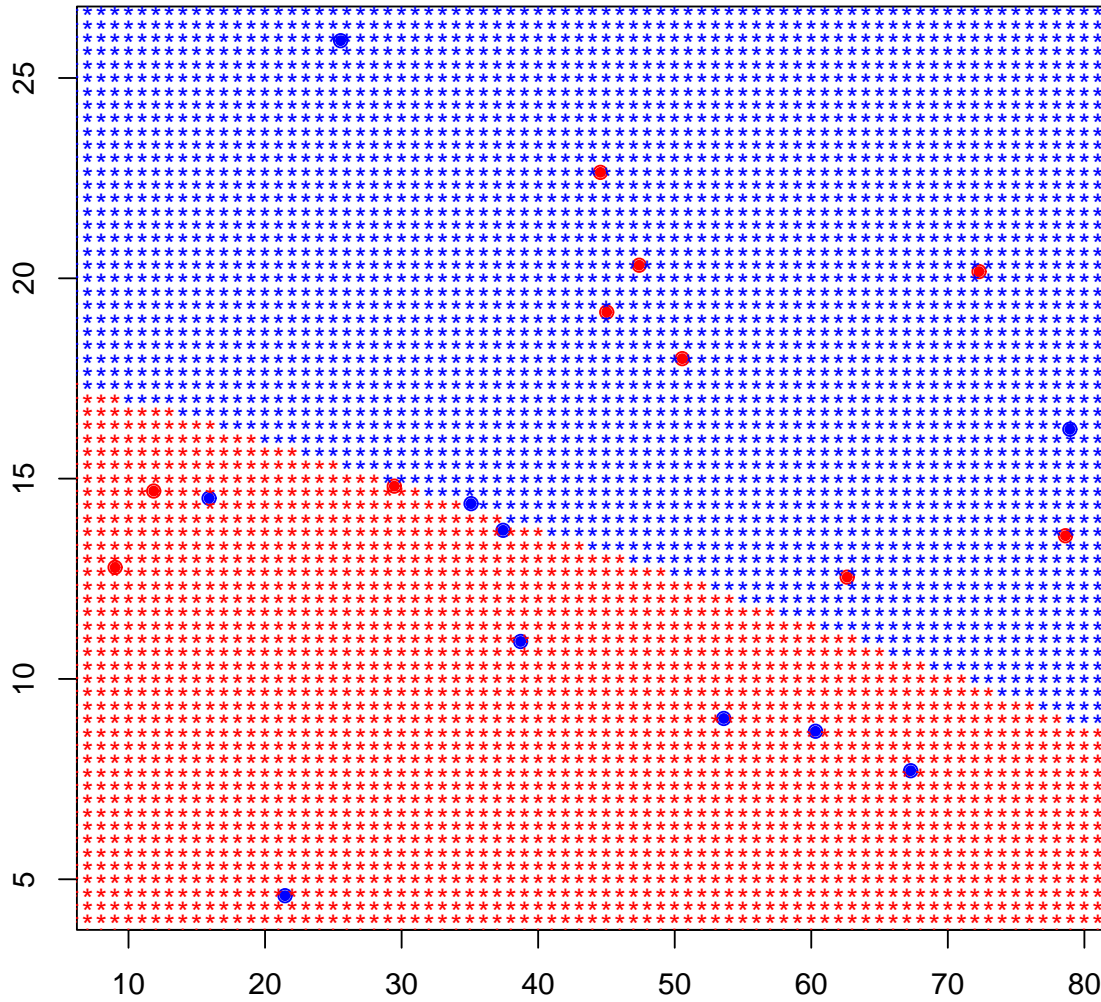
$$z \mapsto \exp\left(-\frac{\|x - y\|^2}{10}\right)$$



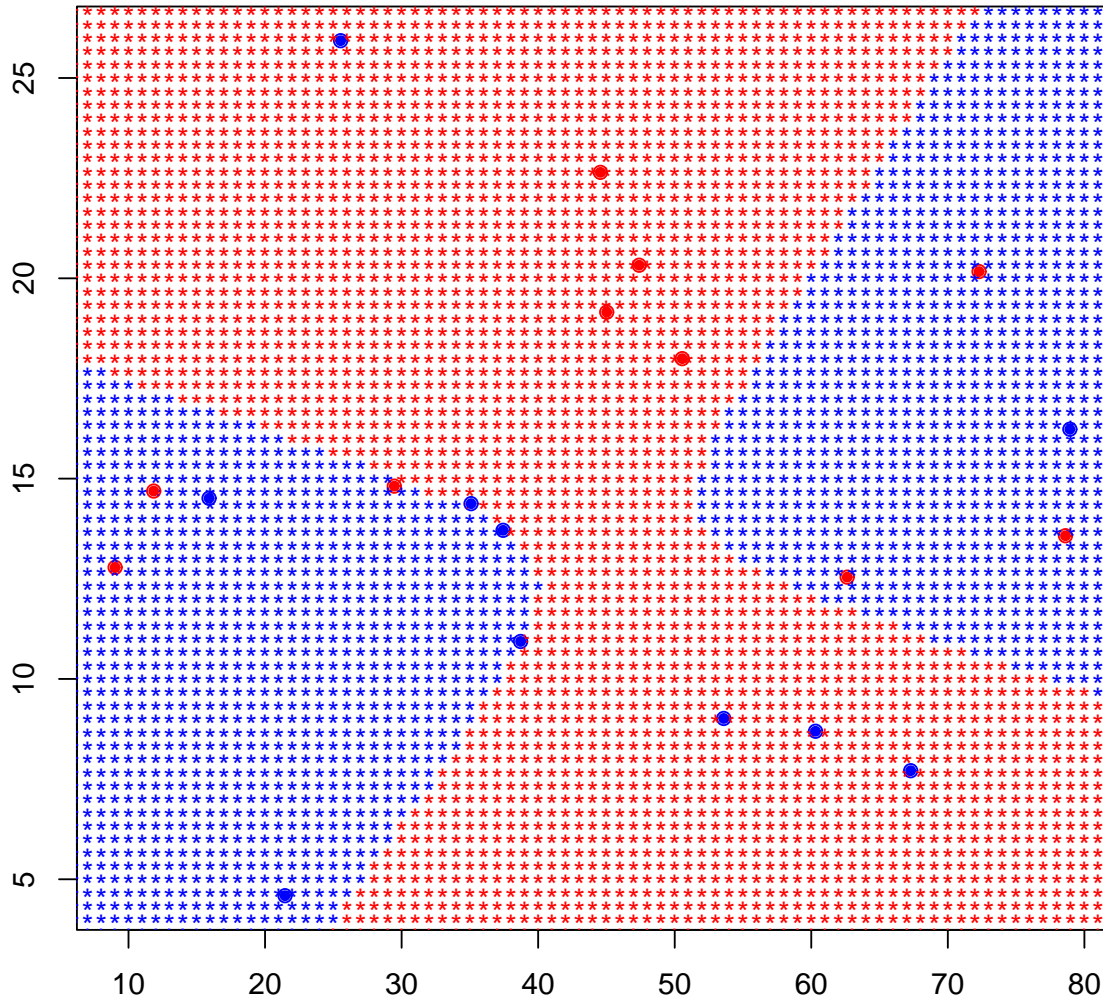
# linear kernel



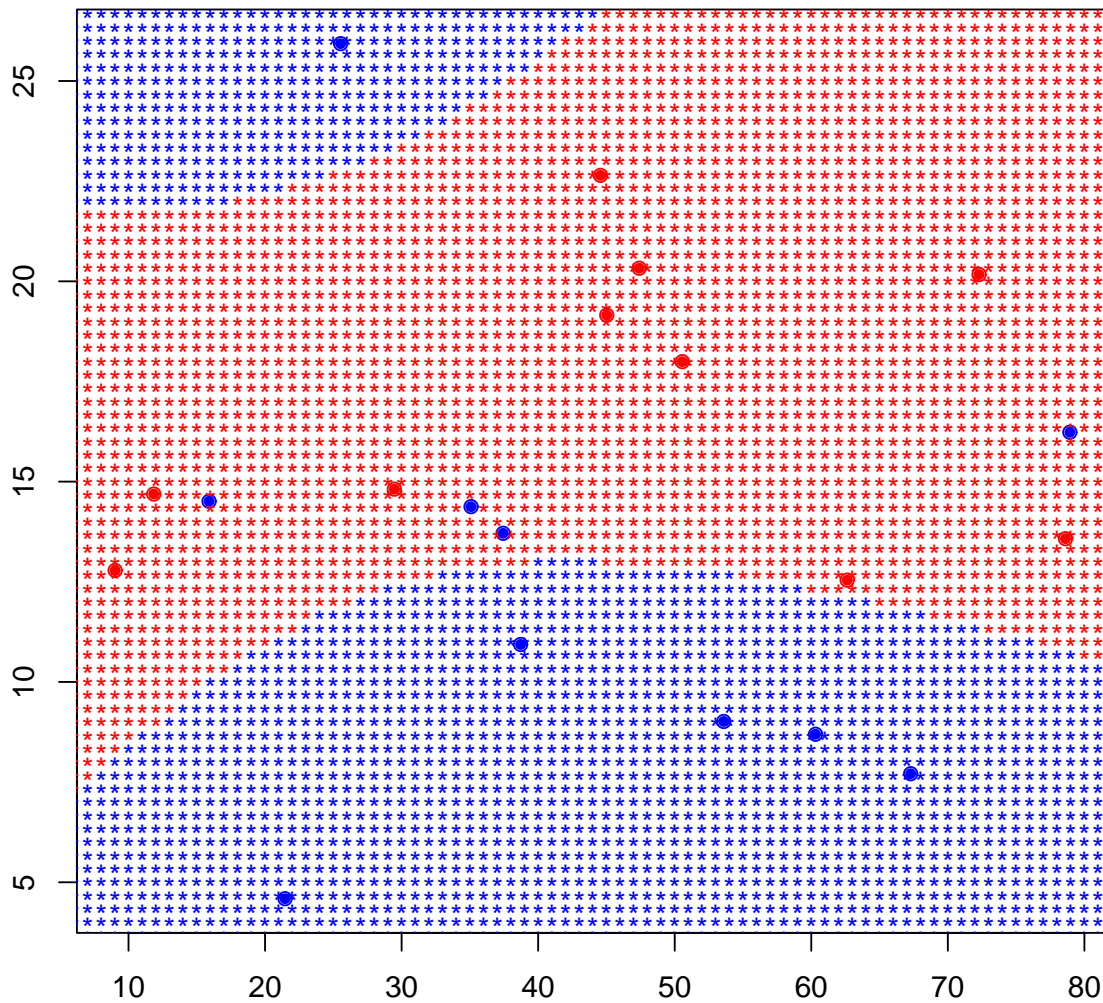
polynomial kernel, degree=2, coef0=-5, gamma=1



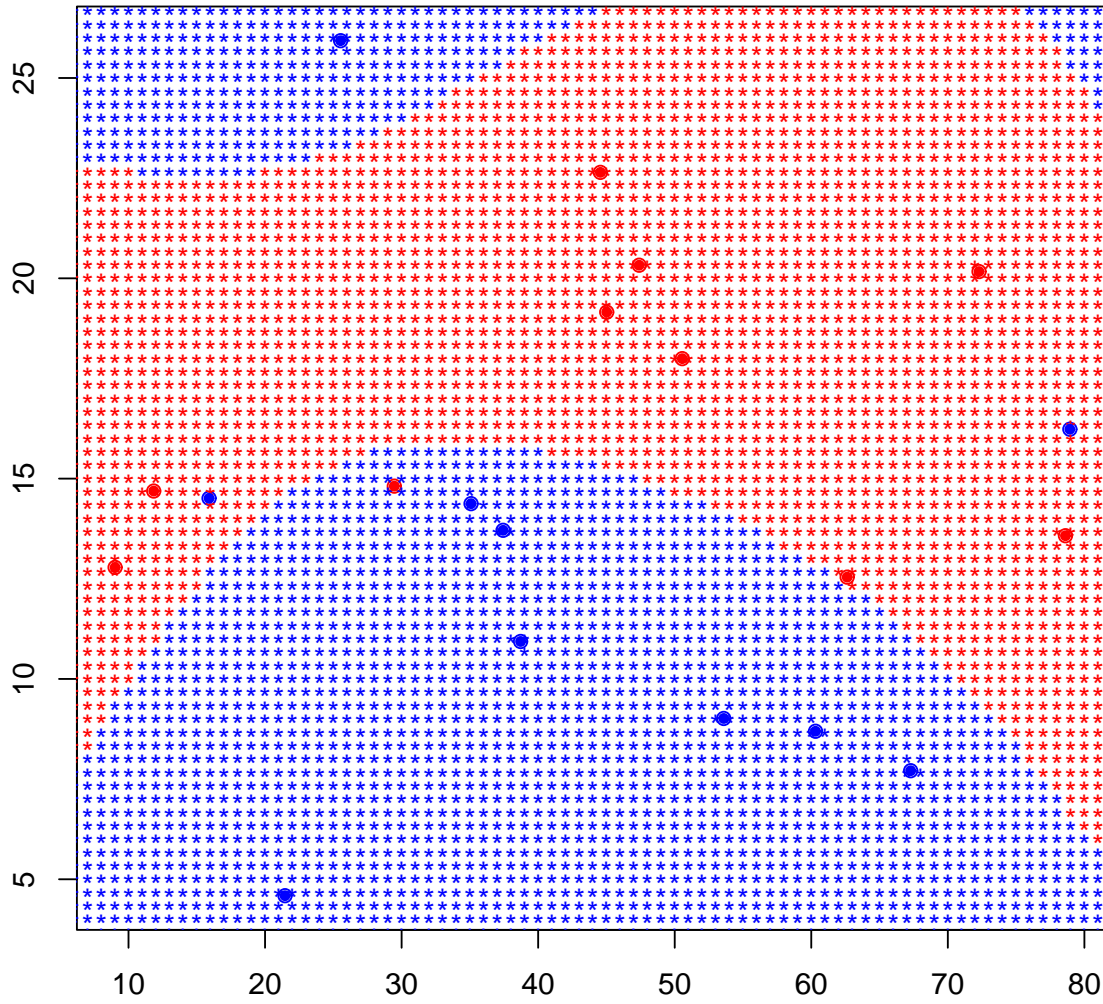
polynomial kernel, degree=5, coef0=-5, gamma=1



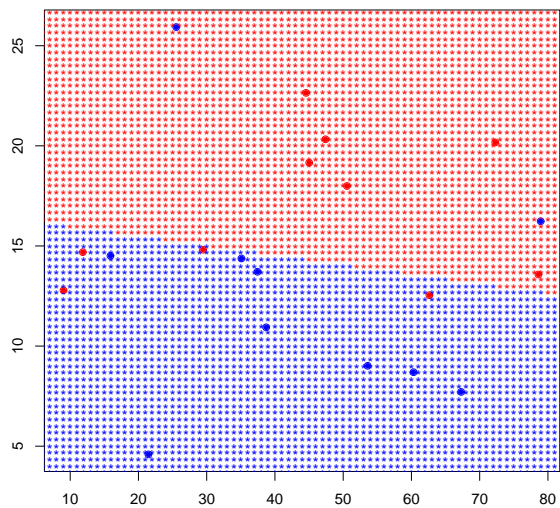
sigmoid kernel, gamma=0.1,coef0=-1



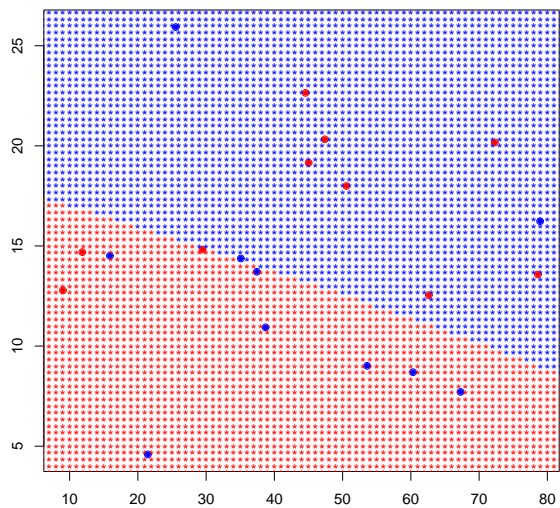
radial kernel, gamma=0.1



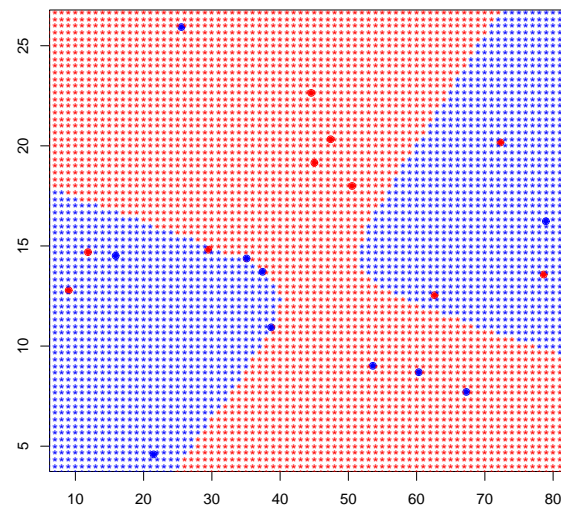
linear kernel



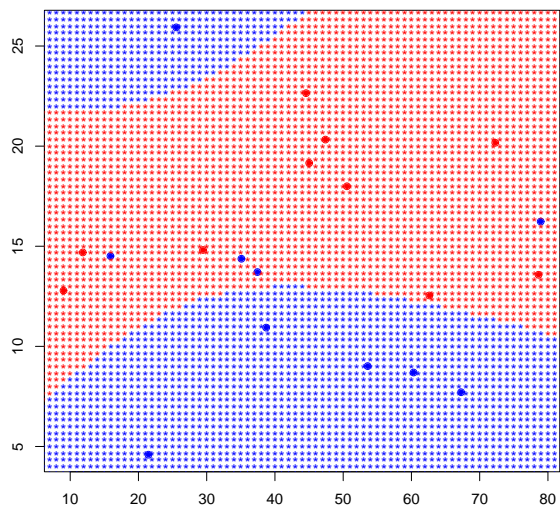
polynomial kernel, degree=2, coef0=-5, gamma=1



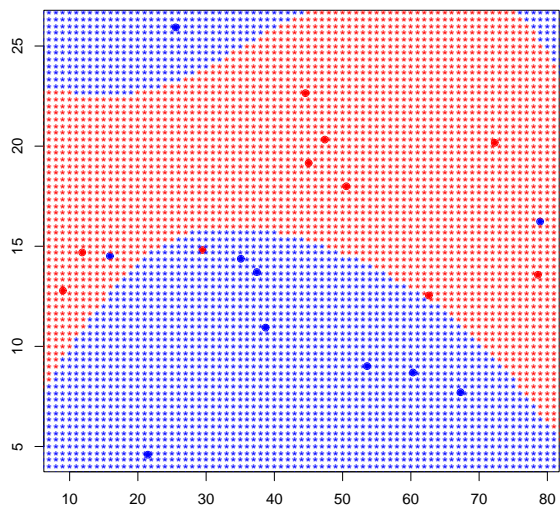
polynomial kernel, degree=5, coef0=-5, gamma=1



sigmoid kernel, gamma=0.1,coef0=-1



radial kernel, gamma=0.1



beliebte Kernels:

polynomial  $d$ -ten Grades:

$$K(x, z) = (1 + \langle x, z \rangle)^d$$

radiale Basis (Gauß-Kern):

$$K(x, z) = \exp\left(-\frac{\|x - z\|^2}{c}\right)$$

sigmoid:

$$K(x, z) = \tanh(\kappa_1 \langle x, z \rangle + \kappa_2)$$

Wieso sind das Kerne?

z.B. polynomialer Kern vom Grad 2 auf  $\mathbb{R}^2$ :

$$\begin{aligned}K(x, z) &= (1 + \langle x, z \rangle)^2 \\&= (1 + x_1 z_1 + x_2 z_2)^2 \\&= 1 + 2x_1 z_1 + 2x_2 z_2 + (x_1 z_1)^2 + (x_2 z_2)^2 + 2x_1 z_1 x_2 z_2 \\&= \phi_1(x)\phi_1(z) + \phi_2(x)\phi_2(z) + \phi_3(x)\phi_3(z) + \dots + \phi_6(x)\phi_6(z) \\&= \langle \phi(x), \phi(z) \rangle\end{aligned}$$

mit  $\phi(x) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)^T \in \mathbb{R}^6$ .

Welche Funktionen  $K : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  sind Kerne?

notwendige Bedingungen:

$$K(x, z) = \langle \phi(x), \phi(z) \rangle = \langle \phi(z), \phi(x) \rangle = K(z, x)$$

$$K(x, z)^2 = \langle \phi(x), \phi(z) \rangle^2 \leq \langle \phi(x), \phi(x) \rangle \langle \phi(z), \phi(z) \rangle = K(x, x)K(z, z)$$

weitere notwendige Bedingungen?

hinreichende Bedingungen?

Wir nehmen zunächst an: Nur Eingaben aus endlicher Menge

$X = \{x_1, \dots, x_k\} \subset \mathbb{R}^n$  sind erlaubt, sei

$$K : X \times X \rightarrow \mathbb{R}$$

eine symmetrische Abbildung.

Betrachte Matrix

$$M := (K(x_i, x_j))_{i,j=1}^k.$$

Da  $M$  symmetrisch ist, gibt es eine orthogonale Matrix  $V$  mit

$$V^T M V = D$$

so dass  $D$  Diagonalmatrix der Eigenwerte  $\lambda_1, \dots, \lambda_k$  von  $M$  ist, und es gilt

$\lambda_i \in \mathbb{R}$ .

Die Spalten von  $V$  sind die zugeh. Eigenvektoren  $v_1, \dots, v_k$ , und es gilt

$V^T = V^{-1}$  und damit  $M = V D V^T$ .

Wenn  $K$  ein Kern mit Abb.  $\phi$  ist, dann muss gelten:

$$\lambda_i = \sum_{j,l=1}^k V_{ij}^T K(x_j, x_l) V_{li} = \sum_{j,l=1}^k v_{ij} \langle \phi(x_j), \phi(x_l) \rangle v_{il} = \left\langle \sum_{j=1}^k v_{ij} \phi(x_j), \sum_{l=1}^k v_{il} \phi(x_l) \right\rangle \geq 0$$

Umgekehrt: Sind alle  $\lambda_i \geq 0$ , so sei

$$\phi(x_i) := \begin{pmatrix} \sqrt{\lambda_1} v_{1i} \\ \vdots \\ \sqrt{\lambda_k} v_{ki} \end{pmatrix} \in \mathbb{R}^k$$

Dann gilt:

$$\langle \phi(x_i), \phi(x_j) \rangle = \sum_{l=1}^k \lambda_l v_{li} v_{lj} = \sum_{l=1}^k V_{il} D_{ll} V_{lj}^T = M_{ij} = K(x_i, x_j)$$

Wir haben also bewiesen:

**Satz 8** *Ist  $X$  endlich und  $K : X \times X \rightarrow \mathbb{R}$  symmetrisch, so ist  $K$  genau dann ein Kern falls die Matrix  $(K(x, z))_{x, z \in X}$  positiv semidefinit ist.*

*Als Feature-Raum genügt der  $\mathbb{R}^{|X|}$ .*

Nun zu unendlichen Eingabe-Mengen  $X$ , und damit womöglich  $\infty$ -dimensionalen Feature-Räumen  $V$ .

Ein Vektorraum mit einem Skalarprodukt (d.h. symmetrische, positiv definite Bilinearform) heißt **Hilbert-Raum** wenn er separabel ist (d.h. eine in ihm dicht liegende abzählbare Teilmenge besitzt) und vollständig (d.h. jede Cauchy-Folge konvergiert).

Sei  $X \subset \mathbb{R}^n$ . Dann ist  $L_2(X)$  der Hilbert-Raum der stetigen Funktionen  $f : X \rightarrow \mathbb{R}$  mit

$$\|f\|_{L_2} := \int_X f(x)^2 dx < \infty$$

Wir erhalten ein Skalarprodukt auf  $L_2(X)$  durch

$$\langle f, g \rangle := \int_X f(x)g(x)dx \leq \sqrt{\int_X f(x)^2 dx} \sqrt{\int_X g(x)^2 dx} < \infty$$

Für jede Folge  $\lambda = (\lambda_1, \lambda_2, \dots)$  mit  $\forall i : \lambda_i \geq 0$  definieren wir den Hilbert-Raum  $\ell_2(\lambda)$  der Folgen  $(x_1, x_2, \dots)$  mit

$$\sum_{i=1}^{\infty} \lambda_i x_i^2 < \infty$$

mit Skalarprodukt  $\langle x, y \rangle := \sum_i \lambda_i x_i y_i$ .

**Satz 9 (Mercer)** Sei  $X \subset \mathbb{R}^n$  kompakt. Ist  $K : X \times X \rightarrow \mathbb{R}$  stetig und symmetrisch und der Integral-Operator  $T_K : L_2(X) \rightarrow L_2(X)$ ,

$$(T_K f) : y \mapsto \int_X K(y, x) f(x) dx$$

positiv, d.h.  $\forall f \in L_2(X) \int_{X \times X} K(y, x) f(x) f(y) dx dy \geq 0$ , dann gibt es Eigenfunktionen  $\phi_j \in L_2(X)$  von  $T_K$  mit positiven zugehörigen Eigenwerten  $\lambda_j$  mit  $\|\phi_j\|_{L_2} = 1$ , so dass

$$K(x, z) = \sum_{j=1}^{\infty} \lambda_j \phi_j(x) \phi_j(z)$$

**Satz 10**  $\forall f \in L_2(X) \int_{X \times X} K(y, x) f(x) f(y) dx dy \geq 0$  gilt genau dann wenn für jede endliche Teilmenge  $X'$  von  $X$  die Matrix  $(K(x, z))_{x, z \in X'}$  positiv definit ist.

**Beweisidee:**  $f \in L_2(X)$  können beliebig genau durch Treppenfunktionen mit endlich vielen Stufen approximiert werden und umgekehrt (*Details an der Tafel*).



**Satz 11** Seien  $K_1$  und  $K_2$  Kerne auf  $X \times X$ ,  $X \subset \mathbb{R}^n$ ,  $K_3$  ein Kern auf  $\mathbb{R}^m \times \mathbb{R}^m$ ,  $a \in \mathbb{R}_{\geq 0}$ ,  $f : X \rightarrow \mathbb{R}$ ,  $\phi : X \rightarrow \mathbb{R}^m$ ,  $B$  eine symmetrische positiv definite  $n \times n$ -Matrix. Dann sind Kerne:

1.  $K(x, z) = K_1(x, z) + K_2(x, z)$

2.  $K(x, z) = aK_1(x, z)$

3.  $K(x, z) = K_1(x, z)K_2(x, z)$

4.  $K(x, z) = f(x)f(z)$

5.  $K(x, z) = K_3(\phi(x), \phi(z))$

6.  $K(x, z) = x^T Bz$

**Beweis:** Siehe Tafel bzw. Cristianini, Shawe-Taylor (2000), verwende Satz 10.

**Satz 12** Sei  $K_1$  ein Kern über  $X \times X$  und  $p(x)$  ein Polynom mit positiven Koeffizienten und  $c > 0$ . Dann sind Kerne:

1.  $K(x, z) = p(K_1(x, z))$

2.  $K(x, z) = \exp(K_1(x, z))$

3.  $K(x, z) = \exp(-\|x - z\|^2/c)$

**Beweis:** Siehe Tafel bzw. Cristianini, Shawe-Taylor (2000)

1 folgt aus Satz 11,

1  $\Rightarrow$  2 mit  $\exp(x) = \sum_{j=0}^{\infty} \frac{x^j}{j!}$  und Grenzübergang auf Kompaktum  $X$ .

3 folgt aus 1 und 2.

## 4.2 Anwendungsbeispiele aus der Bioinformatik

### 4.2.1 Brown, Grundy, Lin, Cristianini, Sugnet, Ares, Haussler (1999): **Support Vector Machine Classification of Microarray Gene Expression Data**

$m = 79$  cDNA-Microarrays mit  $n = 2487$  Gene von *S. cerevisiae* aus 5 Klassen.  
Aufgabe: entscheide für jedes Protein und jede Klasse ob's dazugehört.

Verglichene Methoden: Lineare Diskriminante, Parzen-Fenster, Entscheidungsbäume, SVM (**Testsieger!**)

“Visual inspection of the raw data indicates that such classification should be possible”

$$X_i := \log \frac{\text{Exprlevel von Gen } x \text{ in Versuch } i}{\text{Exprlevel von Gen } x \text{ in Referenzzustand}}$$

Normalisierung:

$$\bar{X}_i := \frac{X_i}{\sqrt{\sum_{j=1}^m X_j^2}}$$

SVM mit polynomialen Kernen verschiedener Grade und Gauß-Kernen  
(Testsieger!)

Problem: Funktionale Genklassen enthalten sehr wenige Gene, die in den viele negativ-Beispielen untergehen

Lösungsansatz: Ähnlich zu Soft-Margin: Kontrolliere Verhältnis falsch-positive/falsch-negative durch verwendung von  $K(x, x) + \lambda \frac{n^+}{n}$  bzw.  $K(x, x) + \lambda \frac{n^-}{n}$ , wobei  $n^+$  und  $n^-$  die Anzahlen an positiv- und negativ-Beispielen sind.

#### **4.2.2 S. Ramaswamy et al. (2001) Multiclass cancer diagnosis using tumor gene expression signatures. *PNAS* 18(26) 15149-15154**

Affy-Microarrays von 218 Tumor-Proben aus 14 Klassen + 90 Proben gesundes Gewebe; 16063 Gene+ESTs

(Proben ausgewählt aus 314 Tumor- und 98 gesunden Proben nach Qualitätskontrolle)

Vorbehandlung der Daten mit AffyNetrix-Software

Klassifikation nach jeder Tumorklasse mit linearer SVM, dann One-vs.-All (OVA): nimm die Klasse  $k$  für die  $\langle \beta^{(k)}, x \rangle + \beta_0^{(k)}$  möglichst groß wird

(nach Minimierung von  $\|\beta^{(k)}\|^2$  bzgl.  $\forall i : y_i (\langle \beta^{(k)}, x_i \rangle + \beta_0^{(k)}) \geq 1$ ).

Kreuzvalidierung der Methode: von 144 Tumoren wurde jeweils einer ausgelassen, die SVM mit den restlichen trainiert, der eine musste dann vorhergesagt werden.

Außerdem wurde noch mit den 144 trainiert, um dann 54 andere zu klassifizieren.

Daraus geschätzte Korrektheit: 78%

Problem:  $\beta$  “enthält” alle Gene und ist daher schwer zu interpretieren

**Recursive Feature Elimination:** Gene, für die  $|\beta_i|$  in unteren 10%-Bereich ist, werden eliminiert, dann wird OVA-SVM wiederholt; das ganze wird iteriert.