Kerne (2) und die Support Vector Machine

Volker Tresp

Kostenfunktionen

- Bisher hatten wir gezeigt, dass sich für die Lösung der lineare Regression als gewichtete Summe von Kernelfunktionen schreiben lässt
- Für welche weiteren Kostenfunktionen ist dies ebenso möglich? Hier liefert das folgende Theorem Klarheit

Representer Theorem

- *Representer Theorem:* Sei Ω eine strikt monoton wachsende Funktion, loss() eine beliebige Verlustfunktion, dann erlaubt der Minimierer des regularisierten Risikos

$$\sum_{i=1}^{N} \text{loss}(y_i, f(\mathbf{x}_i)) + \Omega(< f, f >_\phi)$$

  eine Darstellung der Form

$$f(\mathbf{x}) = \sum_{i=1}^{N} v_i k(\mathbf{x}_i, \mathbf{x})$$

- Beispiel: $< f, f >_\phi = w^T w$
- Dies bedeutet, dass sich Kernellösungen z.B. auch für die logistische Regression und die *Optimale Hyperebene* ergeben
- Allerdings lassen sich die Lösungen in der Regel nicht geschlossen schreiben sondern ergeben sich als Lösung eines Optimierungsproblems

Support Vector Machine

- In Bezug auf das Prinzip der optimalen Hyperebene ergibt sich die Support Vector Machine
- Wir hatte ja schon gesehen, dass sich die Lösungen entsprechend schreiben lassen

Support Vector Machine (trennbare Klassen)

- Man löst schließlich: Maximiere in Bezug auf die α_i

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k \tilde{\phi}(\mathbf{x}_i)^T \tilde{\phi}(\mathbf{x}_k)$$

$$= \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_k)$$

mit den Nebenbedingungen

$$\alpha_i \geq 0$$

und

$$0 = \sum_{i=1}^{N} \alpha_i y_i$$

- Damit kann die Lösung sich schreiben als

$$f(\mathbf{z}) = \text{sign}\left(\sum_{i=1}^{N} y_i \alpha_i k(\tilde{\mathbf{z}}, \tilde{\mathbf{x}}_i) + w_0\right)$$

  Die Terme mit $\alpha_i > 0$ definieren die Supportvektoren. Dies bedeutet, dass man die Lösung ebenfalls als gewichtete Summe der inneren Produkte des Eingangsvektors mit den Supportvektoren schreiben kann!

Optimal Trennende Hyperebenen: Nicht-trennbare Klassen

- Bei sich überlappenden Klassen führt man *slack* Variablen ξ_i ein:
- Die optimale Trennebene kann gefunden werden als

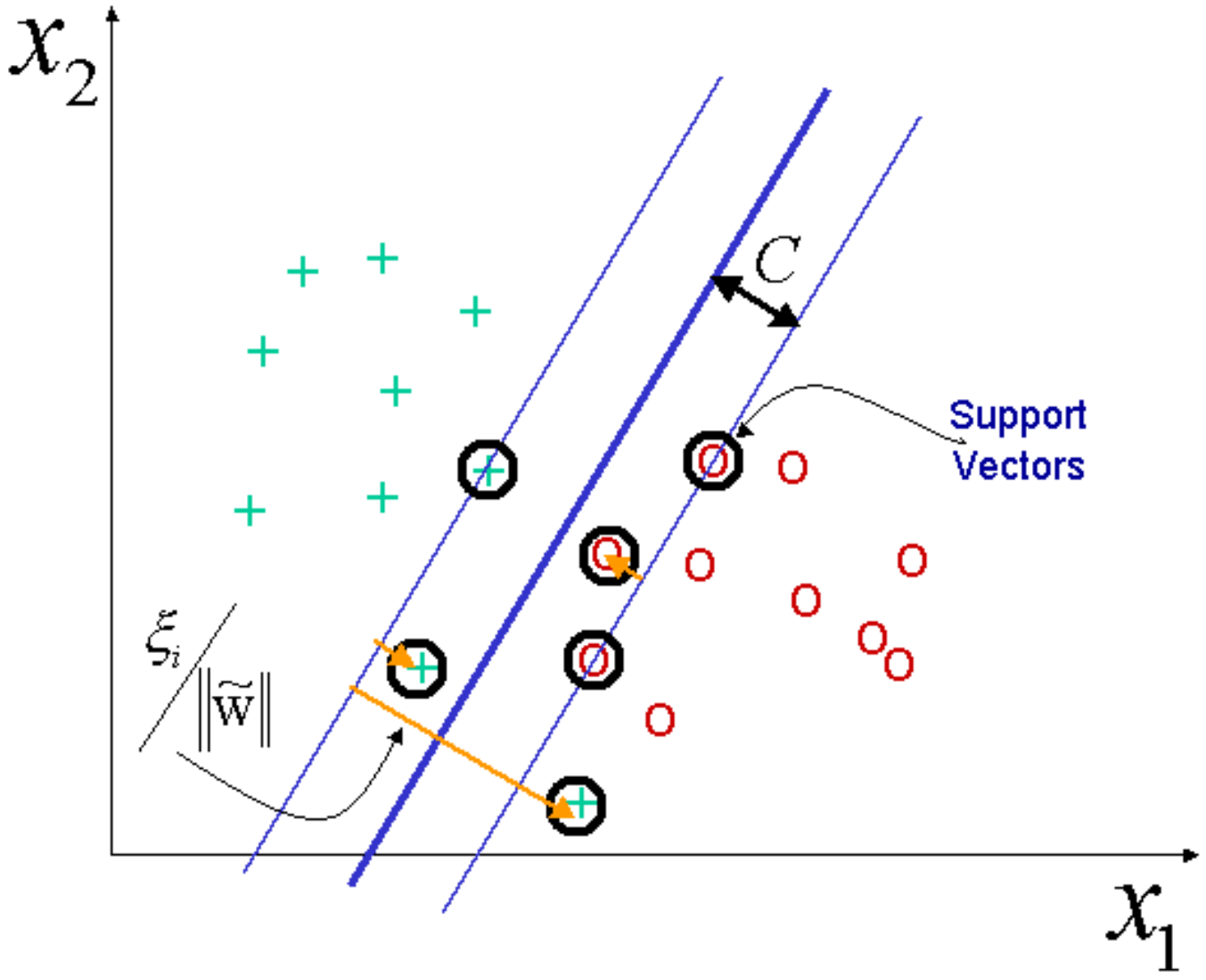
$$\mathbf{w}_{opt} = \arg\min_{\mathbf{w}} \tilde{\mathbf{w}}^T \tilde{\mathbf{w}}$$

unter den Nebenbedingungen, dass

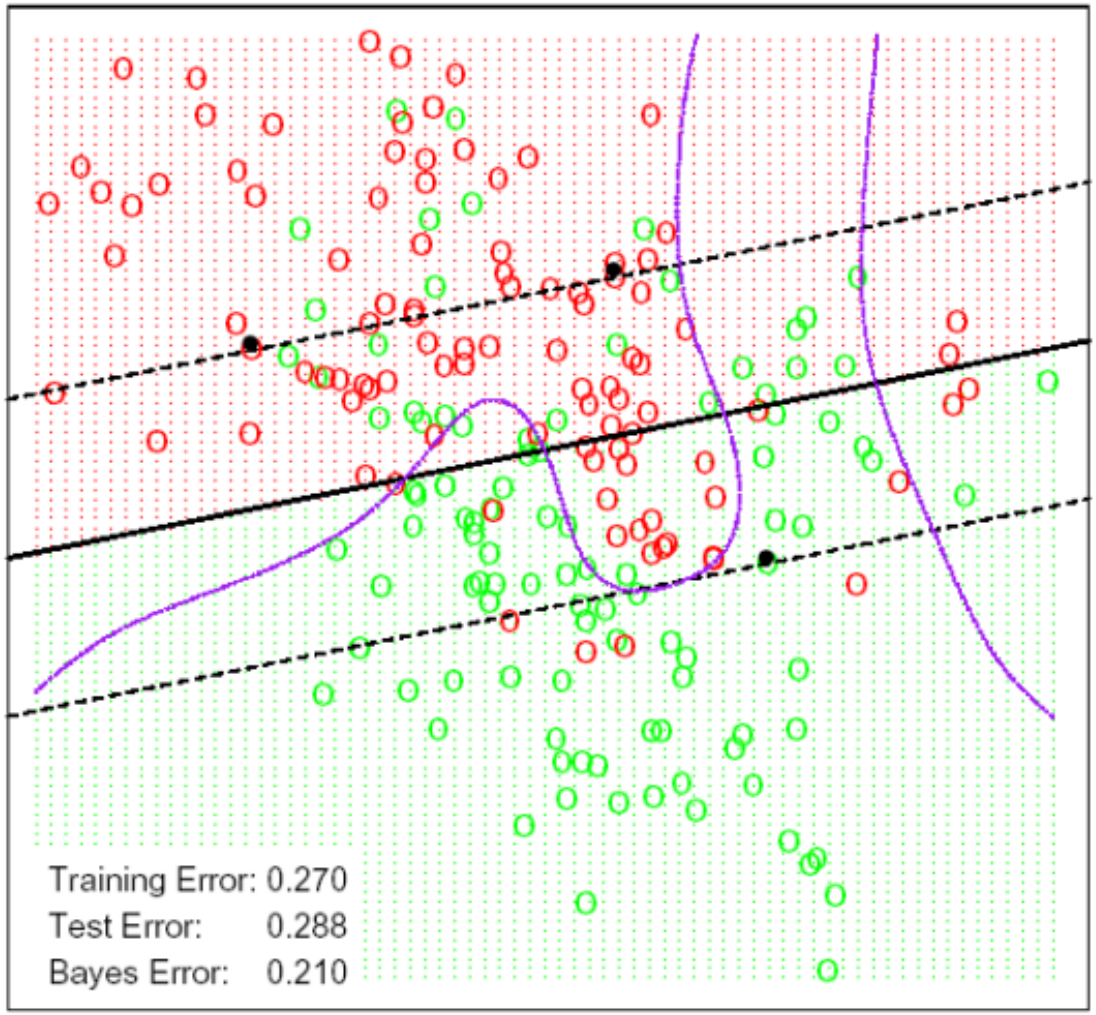
$$y_i(\mathbf{x}_i^T \mathbf{w}) \geq 1 - \xi_i \quad i = 1, \ldots, N$$

$$\xi_i \geq 0 \quad \sum_{i=1}^{N} \xi_i \leq 1/\gamma$$

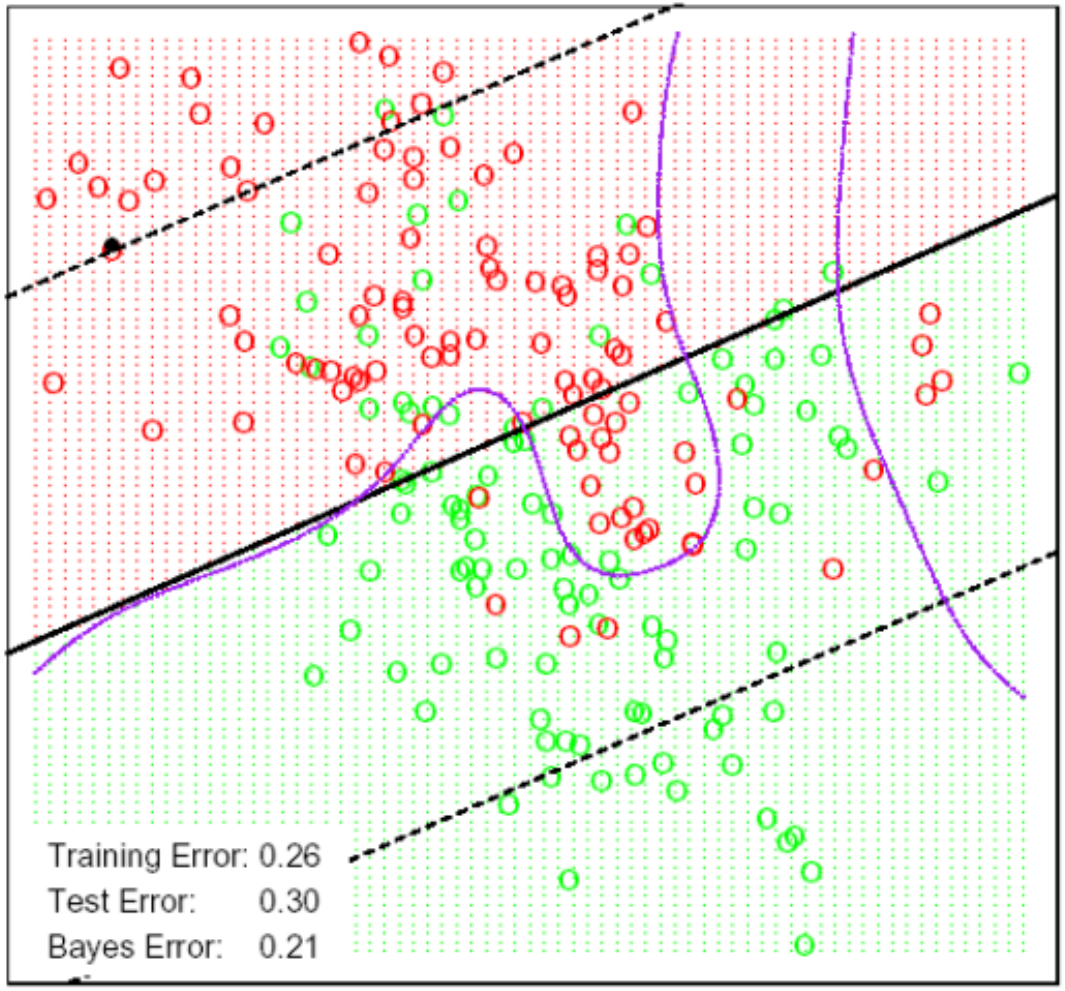
- $\gamma > 0$ bestimmt den Kompromiss zwischen Trennbarbeit von Klassen und Überlappungsgrad. Für $\gamma \longrightarrow \infty$ erhält man den separierbaren Fall.


x_2
C
Support
Vectors
$\frac{\xi_i}{\|\tilde{w}\|}$
x_1

Lineare SVM

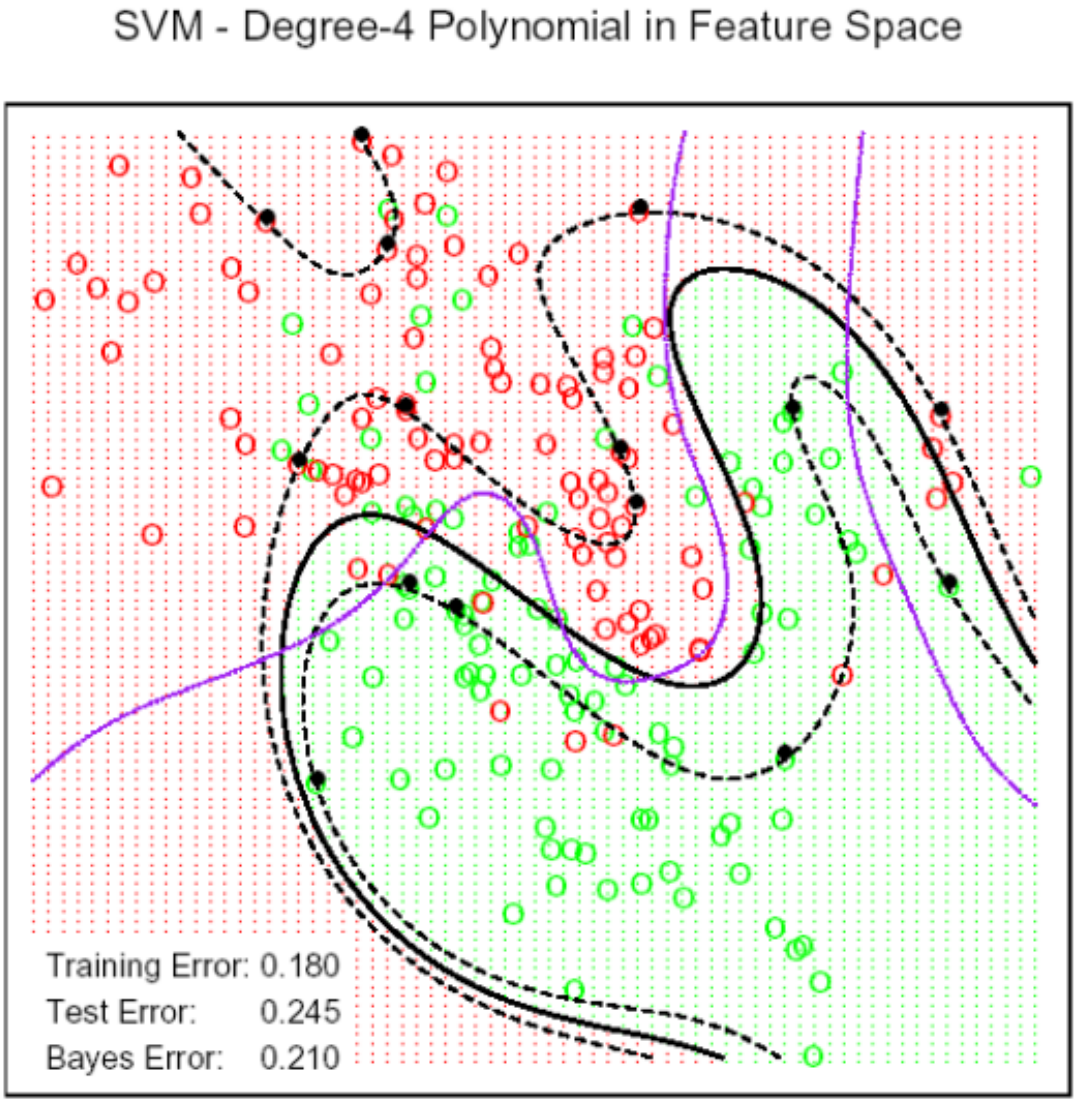


$$\gamma = 10000$$

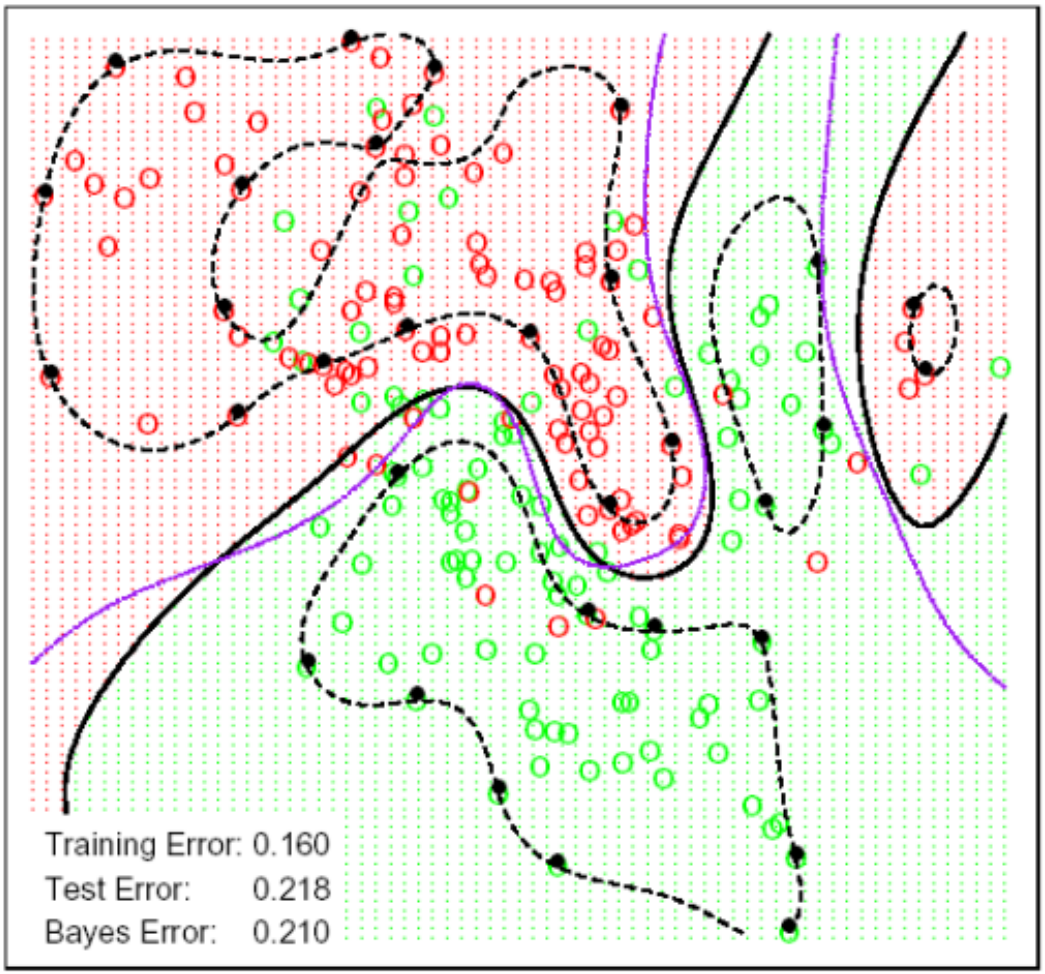


$\gamma = 0.01$

Nichtlineare SVM



SVM - Radial Kernel in Feature Space



Bemerkungen

- Der "Kernel Trick" erlaubt es, in unendlich hohen Dimensionen zu arbeiten
- Dennoch können durch die Regularisierung sehr gute Ergebnisse erzielt werden; z.B. hängt die Generalisierung einer SVM vom Margin ab und nicht von der Dimensionalität des Problems
- Die Regularisierung entspricht Glattheitsannahmen der Funktion; es gibt eine gewisse äquivalenz zwischen Kernel Trick, SVM, Gauss Prozessen, smoothing Splines, Regularisierungsansätzen, ...
- Der Kernel Trick kann immer dann angewandt werden, wenn sich die Lösung so schreiben lässt, dass die Eingangsvektoren als innere Produkte auftreten; das bekannteste Beispiel ist die Kernel-PCA

Appendix: RKHS

- Die Lösungen der Optimierungsaufgabe lässt sich schreiben als

$$f(\mathbf{x}) = \sum_{i=1}^{N} v_i k(\mathbf{x}_i, \mathbf{x})$$

- Man kann nun ein inneres Produkt zwischen Funktionen f und g definieren als

$$< f, g >= \sum_{i=1}^{N^f} \sum_{j=1}^{N^g} v_i^f v_j^g k(\mathbf{x}_i, \mathbf{x}_j) = {w^f}^T w^g$$

  und kann zeigen, dass die Funktionen mit diesem inneren Produkt einen Hilbertraum formen.

$$< f, k(z, .) >= \sum_{i=1}^{N^f} v_i^f k(\mathbf{x}_i, z) = f(z)$$

- Dies ist die reproduzierende Eigenschaft des Kernes. Ein Hilbert-Raum mit einem reproduzierenden Kern nennt man *reproducing kernel Hilbert space* (RKHS)

Kernel PCA

- Wir haben gesehen, dass die Lösung einer Klasse von Optimierungsaufgaben zu Kern-Lösungen führt
- Es gibt weitere Klassen von Problemen, die sich in Kernform schreiben lassen: im Grunde alle Probleme, die sich so schreiben lassen, dass Datenvektoren nur als inneres Produkt auftauchen.
- Betrachtet man die PCA: XX^T ist die Kernmatrix und X^TX die Kovarianzmatrix. Wir haben $X = USV^T$. Daher, $S^{-1}U^TX = V^T$. Die Projektion eines neuen Vektors:

  $$V^Tx = S^{-1}U^TXx = S^{-1}U^Tk(.,x)$$

  Da U und S über die Zerlegung der Kernmatrix berechnen lässt, können wir die PCA nur mit Kernoperationen berechnen.