Kernels: Teil 1

Volker Tresp

Einführung Kerne

- Wir haben mehrere Verfahren zur linearen Regression und Klassifikation kennengelernt
- Wie gesehen können diese Verfahren ebenso leicht in Kombination mit Basisfunktionen angewandt werden; mit dieser Sichtweise sind lineare Funktionen nur spezielle Basisfunktionen
- Gegeben eine Menge von Basisfunktionen lassen sich sogenannte Kernelfunktionen berechnen und es stellt sich heraus, dass sich die optimale Lösung nicht nur als lineare Kombination von Basisfunktionen sondern auch als lineare Kombination von (maximal) N Kernelfunktionen schreiben lässt
- Der entscheidende Vorteil ist, dass man nun auch mit sehr vielen (und sogar unendlich vielen) Basisfunktionen arbeiten kann

Einführung Kerne (2)

- In der Vorlesung über Memory-basierte Systeme haben wir die Wichtigkeit des Ähnlichkeitsmaßes betont
- Die Kernelfunktion entspricht genau diesem Ähnlichkeitsmaß
- Vereinheitlichende Sicht: Linearen Modellen, Modellen aus festen Basisfunktionen und im gewissen Sinne auch neuronalen Netzen können Kernel-abstandsmaße zugeordnet werden

Kernels: Lineare Regression

- Betrachten wir die regularisierte Kostenfunktion für lineare Regression

$$J_N^{pen}(\mathbf{w}) = \sum_{i=1}^{N}(y_i - \mathbf{x}_i^T\mathbf{w})^2 + \lambda \sum_{i=0}^{M} w_i^2 = (\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda \mathbf{w}^T\mathbf{w}$$

- ...und die Ableitung nach den Parametern, die in wir gleich Null setzten

$$\frac{\partial J_N^{pen}(\mathbf{w})}{\partial \mathbf{w}} = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}) + 2\lambda\mathbf{w} \doteq 0$$

- Obwohl dies nicht eine explizite Lösung darstellt, bedeutet dies, dass man schreiben kann

$$\mathbf{w}_{Pen} = \frac{1}{\lambda}\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}_{Pen})$$

Kernels: Lineare Regression (2)

- Noch einmal ...

$$\mathbf{w}_{Pen} = \frac{1}{\lambda}\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}_{Pen}) = \frac{1}{\lambda}\sum_{i=1}^{N}\mathbf{x}_i(y_i - \widehat{y}_i)$$

  Der optimale Gewichtsvektor lässt sich schreiben als **lineare gewichtete Überlagerung der Eingangsvektoren**; die Gewichtung entspricht dem Restfehler

- Das heißt man kann allgemein schreiben

$$\mathbf{w}_{Pen} = \sum_{i=1}^{N}\mathbf{x}_i v_i = \mathbf{X}^T\mathbf{v}$$

Kernels: Lineare Regression (3)

- Dies ist nur eine implizite Definition der Lösung; wir können nun jedoch $\mathbf{w}_{Pen} = \mathbf{X}^T\mathbf{v}$ als Nebenbedingung verwenden und in die Kostenfunktion einsetzten und erhalten

$$J_N^{pen}(\mathbf{v}) = (\mathbf{y} - \mathbf{X}\mathbf{X}^T\mathbf{v})^T(\mathbf{y} - \mathbf{X}\mathbf{X}^T\mathbf{v}) + \lambda\mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v}$$

$$= (\mathbf{y} - K\mathbf{v})^T(\mathbf{y} - K\mathbf{v}) + \lambda\mathbf{v}^T K\mathbf{v}$$

  wobei K eine $N \times N$ Matrix ist mit Elementen

$$k_{i,j} = \mathbf{x}_i^T\mathbf{x}_j \quad \text{und mit} \quad \mathbf{v} = (v_1, \ldots, v_N)^T$$

- Ein weiteres wichtiges Ergebnis: **Das Optimierungsproblem können wir so schreiben, dass nur die inneren Produkte der Eingangsvektoren $\mathbf{x}_i^T\mathbf{x}_j$ auftauchen, aber nicht die Eingangsvektoren selber!**

Kernels: Lineare Regression (4)

- Wir können nun die Kostenfunktion nach $\mathbf{v}$ ableiten (beachte, dass $K = K^T$)

$$\frac{\partial J_N^{pen}(\mathbf{v})}{\partial \mathbf{v}} = 2K(\mathbf{y} - K\mathbf{v}) + 2\lambda K\mathbf{v}$$

So dass

$$\mathbf{v}_{pen} = (K + \lambda I)^{-1}\mathbf{y}$$

und $\mathbf{w}_{Pen} = \mathbf{X}^T(K + \lambda I)^{-1}\mathbf{y}$

Kernels: Lineare Regression (5)

- Die Vorhersage mit Eingang $\mathbf{z}$ wird somit

$$\hat{t} = w^T \mathbf{z} = v^T X \mathbf{z} = \sum_{i=1}^{N} v_i x_i^T \mathbf{z} = \sum_{i=1}^{N} v_i k(x_i, \mathbf{z})$$

- Wieder ein wichtiges Resultat: auch die Vorhersage lässt sich so schreiben, dass nur innere Produkte verwendet werden; **die Lösung läßt sich als gewichtete Summe von N Kernelfunktionen schreiben**.

Beispiel: Dokumentenanalyse

- Die Merkmalsvektoren sind Worthäufigkeiten, die über ihre Länge normiert werden

$$\mathbf{x}_i \leftarrow \frac{\mathbf{x}_i}{|\mathbf{x}_i|}$$

- Das Cosinusmaß entspricht genau dem inneren Produkt der Merkmalsvektoren, also einem linearen Kernel $k(\mathbf{x}_i, \mathbf{z}) = \mathbf{x}_i^T \mathbf{z}_i$
- Bei einem binären Klassifikationsproblem (als Zielwerte +1, -1, Beispiel: Spam / Nicht Spam) ergibt sich
  - Regression: $\widehat{y}(\mathbf{z}) = \mathbf{z}^T w$
  - Kernel: $\widehat{y}(\mathbf{z}) = \sum_{i-1}^{N} k(\mathbf{z}, \mathbf{x}_i) v_i$
  - Nachbarschaftsmodell (Kernelglätter): $\widehat{y}(\mathbf{z}) = \frac{1}{\sum_{i-1}^{N} k(\mathbf{z}, \mathbf{x}_i)} \sum_{i-1}^{N} k(\mathbf{z}, \mathbf{x}_i) y_i$

Kernels: Nichtlineare Regression (1)

- Die Geschichte wird noch interessanter, wenn man bedenkt, dass man auch mit Basisfunktionen arbeiten kann
- Regularisierte Kostenfunktion

$$J_N^{pen}(\mathbf{w}) = \sum_{i=1}^{N}(y_i - \phi(\mathbf{x_i})^T\mathbf{w})^2 + \lambda\sum_{i=0}^{M} w_i^2 = (\mathbf{y}-\mathbf{\Phi w})^T(\mathbf{y}-\mathbf{\Phi w}) + \lambda\mathbf{w}^T\mathbf{w}$$

$$\frac{\partial J_N^{pen}(\mathbf{w})}{\partial \mathbf{w}} = -2\mathbf{\Phi}^T(\mathbf{y} - \mathbf{\Phi w}) + 2\lambda\mathbf{w}$$

So dass,

$$\mathbf{w}_{Pen} = \frac{1}{\lambda}\mathbf{\Phi}^T(\mathbf{y} - \mathbf{\Phi w}_{Pen}) = \mathbf{\Phi}^T\mathbf{v} = \sum_{i=1}^{N} v_i\phi(\mathbf{x_i})$$

Mit

$$v_i = \frac{1}{\lambda}\left(y_i - \phi(\mathbf{x_i})^T\mathbf{w}_{Pen}\right)$$

Kernels: Nichtlineare Regression (2)

- Dies ist nur eine implizite definition der Lösung; wir können nun jedoch dies als Nebenbedingung verwenden und in die Kostenfunktion einsetzten und erhalten

$$J_N^{pen}(\mathbf{v}) = (\mathbf{y} - \mathbf{\Phi}\mathbf{\Phi}^T\mathbf{v})^T(\mathbf{y} - \mathbf{\Phi}\mathbf{\Phi}^T\mathbf{v}) + \lambda\mathbf{v}^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{v}$$

$$= (\mathbf{y} - K\mathbf{v})^T(\mathbf{y} - K\mathbf{v}) + \lambda\mathbf{v}^T K\mathbf{v}$$

  wobei K eine $N \times N$ Matrix ist mit Elementen

$$k_{i,j} = \phi(\mathbf{x_i})^T\phi(\mathbf{x_j})$$

  und mit

$$\mathbf{v} = (v_1, \ldots, v_N)^T$$

Kernels: Nichtlineare Regression (3)

- Wir können nun die Kostenfunktion nach $\mathbf{v}$ ableiten (beachte, dass $K = K^T$)

$$\frac{\partial J_N^{pen}(\mathbf{v})}{\partial \mathbf{v}} = 2K(\mathbf{y} - K\mathbf{v}) + 2\lambda K\mathbf{v}$$

So dass

$$\mathbf{v}_{pen} = (K + \lambda I)^{-1}\mathbf{y}$$

Eine Vorhersage lässt sich somit schreiben als

$$\widehat{f}(\mathbf{z}) = \phi(\mathbf{z})^T\mathbf{w} = \phi(\mathbf{z})^T\mathbf{\Phi}^T\mathbf{v}_{pen} = \sum_{i=1}^{N} v_i k(\mathbf{z}, \mathbf{x}_i)$$

Mit

$$k(\mathbf{z}, \mathbf{x}_i) = \phi(\mathbf{z})^T\phi(\mathbf{x_i})$$

Bemerkungen und Interpretation des Kernels

- Dies ist nun schon interessanter, da es durchaus mehr Basisfunktionen als Eingangsdimensionen geben kann; insbesondere gilt das Resultat auch, wenn man mit **unendlich vielen Basisfunktionen** arbeitet
- Man kann sogar direkt mit den Kernfunktionen arbeiten, ohne sich besondere Gedanken über die Basisfuktionen zu machen, aus denen sie hergeleitet wurden
- Interpretation des Kernels
  - Als inneres Produkt $k(\mathbf{x}_i, \mathbf{z}) = \mathbf{x}^T\mathbf{z}$
  - Als Kovarianz: wie stark sind Funktionswerte an verschiedenen Eingangspunkten korreliert $k(\mathbf{x}_i, \mathbf{z}) = cov(f(\mathbf{x}_i), f(\mathbf{z}))$
- Wenn $N >> M$ ist die originale Formulierung im Merkmalsraum effizienter; wenn $M >> N$, ist die Kernel-Version effizienter; genauer: im Merkmalsraum benötigt man $M^3 + M^2N$ Operationen und mit Kernels benötigt man $N^3 + N^2M$ Operationen. Wenn die Kernel a prior bekannt sind benötigt man N^3 Operationen

- In speziell strukturierten Problemen lassen sich Kernel berechnen, ohne explizit die NM^2 Operationen ausführen zu müssen. Beispiel: String Kernel. Dies ist eines der wichtigsten aktiven Forschungsaufgaben!

- Dennoch sind nicht alle Funktionen geeignete Kernfunktionen; dies stellt das folgende Theorem dar ...

Mercer Theorem

- Nach Vapnik: The nature of statistical learning theory
- *Mercer Theorem:* Um zu garantieren, dass die symmetrische Funktion $k(\mathbf{x}, \mathbf{z})$ aus L_2 eine Entwicklung der Art

$$k(\mathbf{x}, \mathbf{z}) = \sum_{h=1}^{\infty} \lambda_h \phi_h^T(\mathbf{x}) \phi_h(\mathbf{z})$$

besitzt, mit positiven Koeffizienten $\lambda_h > 0$, so ist es notwendig und ausreichend, dass

$$\int \int k(\mathbf{x}, \mathbf{z}) g(\mathbf{x}) g(\mathbf{z}) d\mathbf{x} d\mathbf{z} > 0$$

gültig ist für alle $g \neq 0$ für welche

$$\int g^2(\mathbf{x}) d\mathbf{x} < \infty$$

- Das Theorem sagt aus, dass für sogenannte positiv-definite Kernels, eine Zerlegung in Basisfunktionen möglich ist!

Kernel Design

- Wir haben bereits lineare Kernel kennengelernt mit

$$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$$

Hier sind sowohl Basisfunktionen als auch die entsprechenden Kernels lineare Funktionen

- Polynomialer Kernel

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j)^d$$

Hier sind die entsprechenden Basisfunktionen alle geordneten Polynome des Grades d

- Polynomialer Kernel (2)

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + R)^d$$

Hier sind die entsprechenden Basisfunktionen alle geordneten Polynome vom Grad d oder kleiner

- Gauss-Kernel (RBF-Kernel)

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{2s^2}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$$

  Diese Kernels entsprechen *unendlich vielen* gaussförmingen Basisfunktionen

- Sigmoider Kernel

$$k(\mathbf{x}_i, \mathbf{x}_j) = \text{sig}\left(\mathbf{x}_i^T \mathbf{x}_j\right)$$

  Interessant im Zusammenhang mit Neuronalen Netze