Kernfunktionen für Strukturierte Daten

Thomas Gärtner

Fraunhofer AIS, Schloß Birlinghoven, 53754 Sankt Augustin
Thomas.Gaertner@ais.fraunhofer.de

Abstract: Maschinelles Lernen ist ein Teilgebiet der Informatik, das sich mit der Automatisierung von Lernprozessen beschäftigt. Kernmethoden — die zur Zeit wohl erfolgreichsten maschinellen Lernverfahren — wurden zunächst für Anwendungen konzipiert, in denen die Objekte des Lernens einfach in einen Euklidischen Raum eingebettet werden können. In vielen Anwendungen ist dies jedoch nicht der Fall.

Diese Arbeit erweitert nun Kernmethoden auf allgemeinere, so genannte strukturierte, Daten. Dazu werden passende Kernfunktionen definiert und charakterisiert. Anwendungen aus dem Bereich der pharmazeutischen Wirkstoffforschung zeigen substantielle Verbesserungen gegenüber dem Einsatz konventioneller Lernverfahren.

1 Einleitung

Die Fähigkeit aus Erlebtem und Beobachtetem zu lernen, also aus äußeren Eindrücken abstrakte Erfahrungen abzuleiten, ermöglicht dem Menschen wie auch autonomen Maschinen die Anpassung an sich ständig verändernde Umgebungen. Je mehr Daten Tag für Tag von Maschinen gesammelt, verarbeitet und gespeichert werden, desto wichtiger wird es, die Analyse der Daten — also das Lernen aus den Daten — zu automatisieren. Dies ist das Ziel des maschinellen Lernens, einem Teilgebiet der Informatik mit Berührungspunkten zu vielen anderen Wissenschaften, wie zum Beispiel der Statistik. Da die Menge der gesammelten Daten in sehr vielen Wissenschafts- und Ingenieurszweigen die manuelle Untersuchung verhindert, wird das maschinelle Lernen mehr und mehr zu einer Schlüsseltechnologie, die andere Technologien verbessert oder sogar erst ermöglicht. Einige Probleme, in denen maschinelle Lernverfahren erfolgreich eingesetzt werden, sind Handschriftenerkennung, Erkennung ungewöhnlicher Ereignisse in Netzwerken, sowie Entdeckung von Kreditkartenmissbrauch. Desweiteren stehen maschinelle Lernverfahren hinter einigen der bedeutendsten, jüngeren technologischen Entwicklungen wie der Suche im WWW, dem rechnergestützten Sehen oder den autonomen Fahrzeugen.

Kernmethoden sind zur Zeit die wohl erfolgreichste Klasse von maschinellen Lernverfahren. Viele Kernmethoden zeichnen sich sowohl durch gute theoretische Grundlagen als auch sehr gute empirische Lernergebnisse aus. Ebenso wie bei klassischen maschinellen Lernverfahren konnten diese Lernerfolge hauptsächlich in Domänen erzielt werden, in denen die Objekte des Lernens recht einfach in einem Euklidischen Raum eingebettet werden können. In vielen Anwendungen ist dies jedoch nicht der Fall. Die Aktivität chemischer Wirkstoffe gegen bestimmte Krankheiten zu schätzen, ist eine derartige An-

wendung. Hierbei ist der Strukturgraph eine natürliche Darstellungsform der Wirkstoffe. Kernmethoden und andere konventionelle maschinelle Lernverfahren können nicht direkt auf Probleme dieser Art angewandt werden.

Diese Arbeit erweitert nun Kernmethoden auf allgemeinere, so genannte strukturierte, Daten. Der natürlichste Ansatz, Kernmethoden auf strukturierte Daten anzuwenden, ist es, eine positiv definite Kernfunktion auf der Menge der möglichen Objektbeschreibungen zu definieren. In dieser Arbeit werden daher passende Kernfunktionen definiert und charakterisiert. Insbesondere betrachten wir zwei Darstellungsarten strukturierter Daten: Logik und Graphen. Als Logik-basierte Darstellung verwenden wir Basisterme einer Logik höherer Ordnung mit polymorphen Typen. Als Graphen betrachten wir gefärbte gerichtete und ungerichtete Graphen. Diese Beschreibungssprachen so zu unterscheiden vereinfacht die Darstellung der Kernfunktionen und bringt Vorteile im Bezug auf den Berechnungsaufwand. Beide Beschreibungssprachen können auf natürliche Weise in bestimmten Anwendungen eingesetzt werden und decken zusammen die meisten — wenn nicht sogar alle — Arten strukturierter Daten ab.

Ein motivierendes Beispiel eines Anwendungsgebietes der in dieser Arbeit vorgestellten Kernfunktionen ist in der Tat das Schätzen sowohl der pharmazeutischen Wirksamkeit chemischer Substanzen gegen bestimmte Krankheiten alsauch deren Nebenwirkungen. Dies ist eine Aufgabe des so genannten überwachten Lernens, bei dem das Ziel ist, aus einer gegebenen Menge von Objekten mit beobachteter Zieleigenschaft, eine Funktion abzuleiten, die in der Lage ist, diese Zieleigenschaft für weitere Objekte (die der selben Verteilung folgen) zu schätzen.

Die hier zusammengefaßte Arbeit [Gär05] ist der erste systematische Ansatz, Kernfunktionen für strukturierte Daten zu definieren und in Lernproblemen mit großen Datensätzen anzuwenden. Wir definieren und charakterisieren geeignete Kernfunktionen und untersuchen den jeweiligen Berechnungsaufwand. Unsere empirischen Vergleiche zeigen, dass Kernmethoden mit unseren Kernfunktionen für strukturierte Daten in diversen Anwendungen wesentlich bessere Vorhersagen erzielen als konventionelle Verfahren.

Diese Zusammenfassung ist wie folgt gegliedert: Zunächst führt Abschnitt 2 kurz in den Kontext des überwachten maschinellen Lernens mit Kernmethoden ein. Dann beschreibt Abschnitt 3 den gewählten Ansatz und die Beiträge dieser Arbeit. Abschnitt 4 fasst einige Anwendungen dieser Arbeit in der pharmazeutischen Wirkstoffforschung zusammen. Schließlich verweist Abschnitt 5 auf mögliche zukünftige Arbeiten.

2 Kernmethoden

Die übliche Problemstellung beim *überwachten Lernen* betrachtet eine Instanzmenge $\mathcal{X}$ und eine Kennzeichnungsmenge $\mathcal{Y}$. Ein typisches Beispiel ist die (binäre) *Klassifikation*, bei der $\mathcal{Y} = \{\top, \bot\}$ ist, und die *Regression*, bei der $\mathcal{Y} = \mathbb{R}$ ist. Lernbeispiele entstammen einer gemeinsamen, beliebigen aber festen Wahrscheinlichkeitsverteilung P auf $\mathcal{X} \times \mathcal{Y}$. Das Lernproblem ist es — gegeben einer endlichen Menge $\{(x_i, y_i)\}_{i=1}^n \subseteq \mathcal{X} \times \mathcal{Y}$, die gemäß P beobachtet wurde — eine Funktion zu finden, die die Kennzeichen neuer Bei-

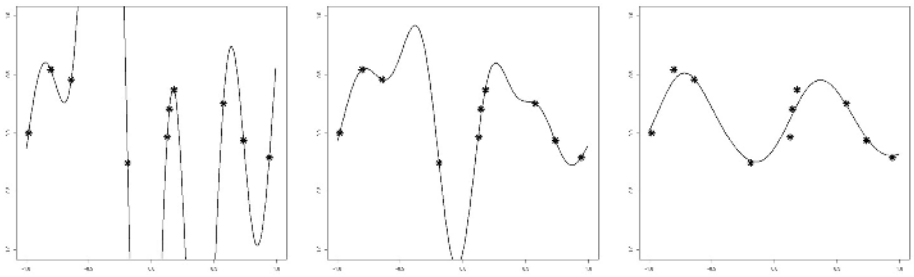

Abbildung 1: Gegeben die Instanzmenge $\mathcal{X} = \mathbb{R}$ (x-Achse), die Kennzeichnungsmenge $\mathcal{Y} = \mathbb{R}$ (y-Achse) und die Trainingsdaten die durch '$*$' dargestellt sind, ist links die Funktion mit minimalem empirischen Fehler dargestellt. Die mittlere und rechte Abbildung zeigen Minima des regularisierten Fehlerfunktionals für unterschiedliche Werte des Parameters ν.

spiele mit geringem Fehler schätzt. Gegeben einem Hypothesenraum $\mathcal{F} \subseteq \{f(\cdot) \mid f : \mathcal{X} \to \mathcal{Y}\}$ ist das Ziel also ein Minimum der Funktion

$$R[f(\cdot)] = \int_{\mathcal{X}\times\mathcal{Y}} V(y, f(x))dP(x, y) \tag{1}$$

zu schätzen ohne P selbst zu kennen, wobei V die Abweichung zwischen beobachteter Kennzeichnung und vorhergesagter Kennzeichnung angibt. Eine typische Fehlerfunktion für Regressionsprobleme ($\mathcal{Y} = \mathbb{R}$) ist der quadratische Fehler $V(y, \hat{y}) = (y - \hat{y})^2$.

Ein üblicher Ansatz hierfür ist es, den *empirischen Fehler* zu minimieren

$$R_{\text{emp}}[f(\cdot)] = \sum_{i=1}^{n} \frac{1}{n} V(y_i, f(x_i)) \; . \tag{2}$$

Dies führt jedoch bei Daten, die verrauscht sind, zu einer Überanpassung des Rauschens, also zu nicht genügend gutem Vorhersagefehler auf neuen Daten. Um eine solche Überanpassung zu vermeiden, beschränken Kernmethoden den Hypothesenraum auf einen Hilbertraum $\mathcal{H}$ und verwenden einen Spezialfall der *Tikhonov Regularisierung* [Tik63]. Es ist dann die Summe aus empirischem Fehler und quadratischer Norm der Funktion im entsprechenden Hilbertraum zu minimieren.

$$\min_{f(\cdot)\in\mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} V(y_i, f(x_i)) + \nu \|f(\cdot)\|^2_{\mathcal{H}} \; . \tag{3}$$

Abbildung 1 vergleicht die Lösung mit minimalem empirischen Fehler mit einer regularisierten Lösung des gleichen Lernproblems.

Der *Repräsentationssatz* [Wah90] besagt nun, dass (für konvexe $V(y, \cdot)$) die Lösungen von (3) die folgende Form haben

$$f^*(\cdot) = \sum_{i=1}^{n} c_i k(x_i, \cdot) \tag{4}$$

wobei $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ der reproduzierende Kern des Hilbertraums $\mathcal{H}$ ist. Die Klasse der reproduzierenden Kernfunktionen entspricht nun genau der Klasse der *positiv definiten Kernfunktionen*. Genau diesen Kernfunktion k entspricht desweiteren eine Abbildung $\phi : \mathcal{X} \rightarrow \mathcal{H}$ so, dass das innere Produkt der Abbildungen gleich dem Wert der Kernfunktion ist, d.h. $\forall\, x, x' \in \mathcal{X} : k(x, x') = \langle \phi(x), \phi(x') \rangle$.

3 Kernfunktionen für Strukturierte Daten

Allgemeine Vorschläge und Prinzipien für Kernfunktionen auf strukturierten Daten finden sich bereits in [Hau99, Wat99]. Beide Arbeiten lassen allerdings — aufgrund der Allgemeinheit der Betrachtung — wichtige Fragen offen und damit dem Anwender überlassen. Andere Arbeiten schlagen dagegen sehr anwendungsspezifische Kernfunktionen vor, die sich nicht leicht auf andere Anwendungsfälle übertragen lassen. Einen Überblick über solche Kernfunktionen gibt [Gär03]. Die vorliegende Arbeit ist nun der erste systematische Ansatz, Kernfunktionen auf strukturierten Daten so zu definieren, dass sich die Form der Kernfunktion aus der Datenstruktur selbst ableiten lässt. Unsere empirischen Vergleiche zeigen, dass Kernmethoden mit unseren Kernfunktionen für strukturierte Daten in diversen Anwendungen wesentlich bessere Vorhersagen erzielen als konventionelle Lernverfahren.

3.1 Kernfunktionen für Basisterme

Zunächst wollen wir Kernfunktionen untersuchen, deren Form aus der Struktur der Lernbeispiele abgeleitet werden kann. Zur Darstellung der Lernbeispiele benötigen wir daher eine mächtige Repräsentationssprache. Hierfür bietet sich eine kürzlich von Lloyd vorgestellte Logik [Llo03] an, die es im Gegensatz zu anderen im maschinellen Lernen verwendeten logik-basierten Repräsentationssprachen erlaubt, Mengen auf natürliche Art und Weise in einem Term darzustellen. Die polymorphen Typen dieser Logik stellen eine ausgezeichnete Struktur dar, anhand derer die Form der Kernfunktion definiert werden kann. Schließlich ist die Darstellung von Lernbeispielen durch abgeschlossene Terme eine sinnvolle Verallgemeinerung der üblichen Attribut-Wert Darstellung.

Die Typen dieser Logik sind aus *Typkonstruktoren*, *Funktionstypen* und *Produkttypen* aufgebaut. Hier dienen Funktionstypen als Typen von Mengen, Multimengen, etc; während Produkttypen als Typen von Tupeln fester Länge dienen. Typkonstruktoren werden dagegen als Typen von Objekten beliebiger Länge und Struktur verwendet, wie zum Beispiel von Sequenzen oder Bäumen. Nicht zuletzt umfasst die Menge der Typkonstruktoren auch Typen für verschiedene Klassen von Zahlen und für anwendungsspezifische Objekte.

Um zum Beispiel den Typ der Teilmengen einer beliebigen Grundmenge zu definieren muss zunächst der Typ der Grundmenge deklariert werden. Dann kann der Teilmengentyp als Funktionstyp deklariert werden, der von dem Typ der Grundmenge auf einen Booleschen Typ abbildet. Die Darstellung einer Teilmenge ist dann die charakteristische Funktion, die genau die Elemente der Teilmenge auf ‘wahr’ abbildet und alle anderen Elemente

der Grundmenge auf 'falsch'.

Basisterme sind eine ausgezeichnete Menge von geschlossenen Termen eines bestimmten Typs, die eindeutig je eine Äquivalenzklasse von Termen des typisierten λ-Kalkulus vertreten und in der üblichen Art und Weise durch Abstraktion, Tupelformung und Applikation geformt werden. Die Basisterme s, t der Menge $\{1, 2\}$ und der Multimenge die aus 42 A, 21 B und keinen weiteren Elementen besteht, sind:

$$s = \lambda x.\mathit{if}\ x = 1\ \mathit{then}\ \top\ \mathit{else}\ \mathit{if}\ x = 2\ \mathit{then}\ \top\ \mathit{else}\ \bot$$
$$t = \lambda x.\mathit{if}\ x = A\ \mathit{then}\ 42\ \mathit{else}\ \mathit{if}\ x = B\ \mathit{then}\ 21\ \mathit{else}\ 0$$

Bevor wir nun die Kernfunktionen für Basisterme definieren können, benötigen wir noch einige Notationen. Bei einer Basisabstraktion r bezeichnet $V(r\ u)$ den Wert von r angewandt auf u, d.h., $V(s\ 2) = \top$ und $V(t\ C) = 0$. Ein Standardwert eines Funktionstyps ist der Wert der Abstraktionen des Typs den diese ohne Bedingung annehmen, d.h., der Standardwert des Typs von s ist $\bot$ und der Standardwert von t ist 0. Die Stütze einer Abstraktion r ist die Menge der Terme u für die $V(r\ u)$ ungleich dem Standardwert ist, d.h., $\mathit{supp}(s) = \{1, 2\}$ und $\mathit{supp}(t) = \{A, B\}$. Weitere Details dieser Logik sind in [Llo03] beschrieben.

Die Kernfunktionen für Basisterme [GLF04] sind nun wie folgt induktiv auf der Struktur der Basisterme definiert: Sind s, t Basisapplikationen, also Terme des gleichen Typkonstruktors (mit gleichen Parametern), so kann s als $C\ s_1 \ldots s_n$ und t als $D\ t_1 \ldots t_m$ dargestellt werden, wobei C, D Datenkonstruktoren und s_i, t_j Basisterme sind. Die Kernfunktion an s, t ist dann definiert durch

$$k(s,t) = \begin{cases} \kappa_T(C,D) & \text{falls } C \neq D \\ \kappa_T(C,C) + \sum_{i=1}^{n} k(s_i,t_i) & \text{sonst} \end{cases}$$

Sind s, t Basisabstraktionen, also Terme des gleichen Funktionstyps, so ist die Kernfunktion an s, t definiert durch

$$k(s,t) = \sum_{\substack{u \in \mathit{supp}(s) \\ v \in \mathit{supp}(t)}} k(V(s\ u), V(t\ v)) \cdot k(u,v).$$

Sind s, t Basistupel, also Terme des gleichen Produkttyps, so kann s als $(s_1, \ldots, s_n)$ und t als $(t_1, \ldots, t_n)$ dargestellt werden, wobei s_i, t_j Basisterme sind. Die Kernfunktion an s, t ist dann definiert durch

$$k(s,t) = \sum_{i=1}^{n} k(s_i,t_i),$$

In [Gär05] und den entsprechenden Vorarbeiten wie [GLF04] sind desweiteren 'Modifikatoren' beschrieben, die eine bessere Anpassung der Kernfunktion an bestimmte Anwendungen erlauben. Dort wird auch die Lernbarkeit einer bestimmten Klasse von Lernproblemen — so genannter "Mehr-Instanz Probleme" — mit diesen Kernfunktionen und quasipolynomieller Fehlerschranke gezeigt. Aus Platzgründen gehen wir hier nicht weiter auf Details ein. Empirische Ergebnisse fassen wir weiter unten kurz zusammen.

3.2 Kernfunktionen für Graphen

Graphen sind eine der wichtigsten Darstellungsformen für strukturierte Daten in Wissenschaft und Technik. Einige Vorarbeiten haben bereits Kernfunktionen für bestimmte — meist recht einfache — Teilklassen von Graphen, wie z.B. für Bäume oder Sequenzen, vorgeschlagen. In diesem Abschnitt betrachten wir nun Kernfunktionen auf allgemeinen Graphen mit dem Ziel, diese in Anwendungen der pharmazeutischen Wirkstoffforschung einsetzen zu können. Die oben eingeführten Basisterme sind zwar prinzipiell in der Lage Graphen darzustellen, doch wurden aus Gründen der Systematik die hierzu nötigen Mechanismen bei der Kerndefinition nicht betrachtet.

Eine allgemeine Methode Kernfunktionen für strukturierte Daten zu erhalten (siehe z.B. [Hau99, Wat99]), ist es, die Objekte in alle möglichen Teile zu zerlegen und als Kernfunktion zweier Objekte ein Maß der Schnittmenge beider Zerlegungen zu verwenden. In diesem Abschnitt werden wir zunächst zeigen, dass dieses Vorgehen bei Graphen zwangsläufig zu Kernfunktionen führt, die nicht effizient berechnet werden können, d.h. sofern P $\neq$ NP können diese Kernfunktionen nicht in polynomieller Zeit berechnet werden. Daher werden wir alternative Kernfunktionen vorschlagen: Eine weg-basierte Kernfunktion, die allgemein bei ungerichteten Graphen effizient berechnet werden kann, und eine zyklus-basierte Kernfunktion, die nur bei einer bestimmten Klasse von Graphen effizient berechnet werden kann. Details sind in [GFW03, HGW04] beschrieben.

Für eine beliebige Menge M sei $[M]^n = \{m \subseteq M : |m| = n\}$. Ein (ungerichteter) *Graph* ist ein Paar $G = (\mathcal{V}, \mathcal{E}) = (\mathcal{V}(G), \mathcal{E}(G))$ bestehend aus einer Menge von "Knoten" $\mathcal{V}$ und einer Menge von "Kanten" $\mathcal{E} \subseteq [\mathcal{V}]^2$, die diese Knoten "verbinden". Häufig werden wir *gefärbte Graphen* $(\mathcal{V}, \mathcal{E}, l)$ betrachten, bei denen eine Funktion $l : \mathcal{V} \cup \mathcal{E} \to \Sigma$ alle Knoten und/oder Kanten eines Graphen auf ein Element des Alphabets Σ abbildet. Die Adjazenzmatrix $E \in \mathbb{R}^{\mathcal{V}\times\mathcal{V}}$ eines Graphen G hat den Eintrag $E_{uv} = 1 \Leftrightarrow \{u, v\} \in \mathcal{E}$ und sonst $E_{uv} = 0$. Ein Graph $G' = (\mathcal{V}', \mathcal{E}')$ ist ein *Teilgraph* eines Graphen $G = (\mathcal{V}, \mathcal{E})$ wenn $\mathcal{V}' \subseteq \mathcal{V}$ und $\mathcal{E}' \subseteq \mathcal{E} \cap [\mathcal{V}']^2$. Zwei Graphen G, G' heißen *isomorph* wenn zwischen ihren Knotenmengen eine kantenerhaltende Bijektion $f : \mathcal{V} \to \mathcal{V}', \{u, v\} \in \mathcal{E} \Leftrightarrow \{f(u), f(v)\} \in \mathcal{E}'$ existiert, f heißt dann ein Isomorphismus. Ein *Homomorphismus* von einem Graphen G in einen Graphen G' ist eine kantenerhaltende Funktion $g : \mathcal{V} \to \mathcal{V}', \{u, v\} \in \mathcal{E} \Rightarrow \{g(u), g(v)\} \in \mathcal{E}'$. Zwischen gefärbten Graphen müssen Isomorphismen und Homomorphismen desweiteren auch die Farben der Kanten und/oder Knoten erhalten. Wenn ein Isomorphismus zwischen zwei Graphen existiert, schreiben wir auch $G \simeq G'$. Wenn ein Isomorphismus zwischen einem Graphen G und einem Teilgraphen eines anderen Graphen G' existiert, schreiben wir $G \lesssim G'$ Die Menge aller (gefärbten) Graphen modulo Isomorphismus (also eigentlich eine Menge von Äquivalenzklassen) bezeichnen wir mit $\mathcal{G}$ $(\mathcal{G}^l)$; $\mathrm{homo}(G, G')$ bezeichnet die Menge aller Homomorphismen zwischen G und G'. Für zwei Graphen G, G' schreiben wir $G + G' = (\mathcal{V} \cup \mathcal{V}', \mathcal{E} \cup \mathcal{E}')$. Ein *Pfad* der Länge n ist ein Graph $P_n = (\{1, \ldots n\}, \{\{i, i+1\} : 1 \leq i < n\})$. Ein *Zyklus* der Länge n ist ein Graph $C_n = P_n + (\emptyset, \{1, n\})$. Die Menge aller Pfade beziehungsweise Zyklen schreiben wir als $\mathcal{P} = \bigcup_n P_n$ und $\mathcal{C} = \bigcup_n C_n$, die Menge aller gefärbten Pfade beziehungsweise Zyklen als $\mathcal{P}^l$ und $\mathcal{C}^l$. Ein *Pfad* beziehungsweise *Zyklus in* einem Graphen ist ein Teilgraph des

Graphen, der isomorph zu einem Pfad bzw Zyklus ist. Ein *Weg in* einem Graphen ist ein Teilgraph, so dass es einen Homomorphismus zwischen einem Pfad und dem Teilgraphen gibt.

Die bereits oben erwähnte Teilgraph-Kernfunktion kann dann definiert werden als

$$k_H(G,G') = |\{h \in H : h \lesssim G\} \cap \{h \in H : h \lesssim G'\}|$$

wobei wir typischerweise als H eine der Mengen $\mathcal{G}$, $\mathcal{P}$ oder $\mathcal{C}$ wählen. Sofern P $\neq$ NP kann man nun zeigen, dass keine der Funktionen $k_\mathcal{G}(G,G')$, $k_\mathcal{P}(G,G')$ und $k_\mathcal{C}(G,G')$ in polynomieller Zeit in der Anzahl der Knoten von G und G' berechnet werden kann, da ansonsten das Hamiltonpfad Problem in polynomieller Zeit gelöst werden könnte, welches jedoch ein NP-vollständiges Entscheidungsproblem ist. Da sich eine solche Kernfunktion also nicht für praktische Zwecke eignet, wollen wir als nächstes untersuchen, ob es überhaupt möglich ist eine Kernfunktion $k : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$ auf Graphen so zu definieren, dass die Funktion $G \mapsto k(G,\cdot)$, die jeden Graphen $G \in \mathcal{G}$ auf eine Funktion $\mathcal{G} \to \mathbb{R}$ abbildet, injektiv ist. Dies wäre wünschenswert, da es nur dann möglich wäre, überhaupt alle Klassifikationen von Graphen zu erlernen. Allerdings zeigt sich, dass dies nur möglich ist, falls $G \simeq G'$ in polynomieller Zeit in der Anzahl der Knoten von G und G' entschieden werden kann. Da für dieses Problem aber angenommen werden muss, dass es nicht in P ist, müssen wir Kernfunktionen betrachten, die diese Anforderung nicht erfüllen.

Zunächst wollen wir die folgende weg-basierte Kernfunktion betrachten:

$$k_w(G,G') = \sum_{p \in \mathcal{P}^l} \lambda_{|\mathcal{V}(p)|} \left|\mathrm{homo}(p,G)\right| \cdot \left|\mathrm{homo}(p,G')\right| \ .$$

Obgleich $\mathcal{P}^l$ nicht endlich ist, können wir $k_w(G,G')$ für bestimmte $\lambda_n > 0$ in polynomieller Zeit in der Anzahl der Knoten von G und G' berechnen. Dazu benötigen wir den Produktgraphen [IK00] $G \times G'$ von G und G':

$$\begin{aligned}\mathcal{V}(G \times G') &= \{(v,v') \in \mathcal{V} \times \mathcal{V}' : l(v) = l'(v')\} \\ \mathcal{E}(G \times G') &= \{\{(u,u'),(v,v')\} \in [\mathcal{V}(G \times G')]^2 : \\ &\qquad \{u,v\} \in \mathcal{E} \wedge \{u',v'\} \in \mathcal{E}' \wedge (l(u,v) = l'(u',v'))\} \ .\end{aligned}$$

Es kann nun gezeigt werden, dass

$$k_w(G,G') = \sum_{p \in \mathcal{P}} \lambda_{|\mathcal{V}(p)|} \left|\mathrm{homo}(p, G \times G')\right| \ .$$

Diese Summe kann nun für bestimmte $\lambda_n > 0$ in polynomieller Zeit berechnet werden, indem die Eigenwerte der Adjazenzmatrix $E^\times$ von $G \times G'$ manipuliert werden. Insbesondere kann $k_w(G,G')$ zum Beispiel für $\lambda_i = \gamma^i$ mit kleinem γ berechnet werden durch $k_w(G,G') = \mathbf{1}^t \left(\mathbf{I} - \gamma E^\times\right)^{-1} \mathbf{1}$ wobei $\mathbf{1}$ den Vektor $(1,1,\ldots)^t \in \mathbb{R}^{\mathcal{V}(G \times G')}$ und $\mathbf{I} \in \mathbb{R}^{\mathcal{V}(G \times G') \times \mathcal{V}(G \times G')}$ die Einheitsmatrix bezeichnet.

Während diese weg-basierte Kernfunktion in polynomieller Zeit berechnet werden kann, ist der Grad dieses Polynoms für Anwendungen in der pharmazeutischen Wirkstoffforschung allerdings noch groß. Für diese Anwendung brauchen wir daher eine spezielle

Kernfunktion, die auf großen Moleküldatenbanken in annehmbarer Zeit berechnet werden kann. Wir werden dazu die oben definierte Kernfunktion $k_{\mathcal{C}^l}(G, G')$ genauer untersuchen. Insbesondere werden wir nun untersuchen, in wiefern die Menge $c(G) = \{h \in \mathcal{C}^l : h \lesssim G\}$ effizient aufgezählt werden kann. Dies kann nicht in polynomieller Zeit in der Anzahl der Knoten von G und G' möglich sein, da $c(G)$ selbst exponentiell groß sein kann. Desweiteren kann $c(G)$ nicht in polynomieller Zeit in $|c(G)|$ und $|\mathcal{V}|$ berechnet werden. Ansonsten könnte für einen Graphen G mit nur einer Farbe $C_n \in c(G)$ in Zeit polynomiell in $|\mathcal{V}|$ entschieden werden, da hier $|c(G)| \leq |\mathcal{V}|$. Dieses Problem ist aber dem NP-vollständigen Entscheidungsproblem Hamiltonkreis gleichbedeutend. Es ist daher interessant zu untersuchen, ob wir eine Menge $C(G)$ in polynomieller Zeit in $|\mathcal{V}|, |C(G)|$ aufzählen können, aus der wir in wiederum polynomieller Zeit $c(G)$ gewinnen können. Sofern $|C(G)|$ für eine bestimmte Klasse von Graphen durch eine Konstante nach oben beschränkt ist, kann dann $c(G)$ für diese Klasse von Graphen effizient berechnet werden. Eine solche Menge $C(G)$ ist zum Beispiel die Menge der Zyklen in G. Hierfür existiert ein Algorithmus [RT75], der diese Menge so aufzählt, dass die Zeit zwischen der Aufzählung zweier Elemente polynomial in $|\mathcal{V}|$ ist. Im besten Fall ist diese Menge nur so groß wie $c(G)$, im schlechtesten Fall allerdings exponentiell größer.

Desweiteren betrachten wir in [HGW04] noch die maximalen Bäume des Waldes, der durch Entfernen der Kanten, die Teil eines Zyklus sind, entsteht. Aus Platzgründen gehen wir hier nicht weiter auf Details ein.

4 Empirische Ergebnisse

Zunächst wollen wir eine kurze Übersicht über einige Anwendungen der Kernfunktionen für Basisterme geben, danach besprechen wir kurz eine Anwendung der Kernfunktionen für Graphen. Die meisten Anwendungen sind aus Platzgründen auf das Wesentliche vereinfacht dargestellt. Details können in [Gär05] bzw. in den entsprechenden Vorarbeiten nachgelesen werden.

Wirkstoffanalyse anhand der 3D Struktur Die 3D Struktur von Molekülen ist wichtig für deren medizinische und pharmazeutische Wirkung. Allerdings kann ein Molekül — je nach energetischem Zustand — unterschiedliche Formen ("Konformationen") annehmen. Ein Molekül ist für viele Anwendungen dann aktiv, wenn eine dieser Formen bestimmte Eigenschaften erfüllt, die aber meist nicht explizit bekannt sind. Um Kernmethoden auf Moleküle anwenden zu können, die durch Mengen von Formen beschrieben sind, modellieren wir die Daten als Basisterme wie folgt: Eine `Konformation` ist ein Tupel von reellen Zahlen, die die Entfernung von einem bestimmten Punkt des Moleküls in verschiedene fest definierte Richtungen zu der Hülle des Moleküls messen. Ein `Molekül` ist ein Funktionstyp, der eine Menge von Konformationen modelliert, in dem er Instanzen des Typs `Konformation` auf Wahrheitswerte abbildet. Auf einem hierfür typischen Datensatz konnten durch Einsatz von Kernmethoden Klassifikationsergebnisse erzielt werden, die dem Stand der Technik entsprechen. Ein großer Vorteil der Kernmethoden ist aller-

dings auch, dass andere Lernprobleme, wie z.B. Regression, unüberwachtes Lernen oder Transduktion, ohne weiteren Aufwand bearbeitet werden können.

Strukturbestimmung von Diterpenmolekülen Ein wichtiger Schritt in der Bestimmung des chemischen Strukturgraphen (der 2D Struktur) von Molekülen ist die Klassifikation von C^{13} NMR Spektren in chemische Strukturklassen. Ein `Spektrum` ist ein Funktionstyp, der die `Frequenz` eines Ausschlags auf die `Multiplizität` des entsprechenden Kohlenstoffatoms, d.h. die Anzahl der damit verbundenen Wasserstoffatome, abbildet. Die `Frequenz` ist eine reelle Zahl und die `Multiplizität` ist ein Element der Menge $\{1, 2, 3, 4, 0\}$ wobei 0 der Standardwert der `Multiplizität` ist. Auf einem hierfür typischen Datensatz konnte durch Einsatz von Kernmethoden eine deutliche Verbesserung der Klassifikationsergebnisse erzielt werden.

Wirkstoffanalyse anhand der 2D Struktur Zwar ist die 3D Struktur auch bei kleinen Molekülen relevant, doch kann sie oft nur mit hohem Aufwand genau bestimmt werden. Zudem können Berechnungsverfahren nur die Position der Atome im Vakuum bestimmen, nicht jedoch im menschlichen Körper. Daher wird in der pharmazeutischen Wirkstoffforschung häufig nur der chemische Strukturgraph betrachtet. Wir haben weg-basierte und zyklus-basierte Kernfunktionen für Graphen in verschiedenen Experimenten auf einem Datensatz (NCI-HIV), der mehr als 42.000 Moleküle enthält, mit früheren Arbeiten verglichen. Dabei hat sich in einem Wilcoxon-Test herausgestellt, dass Kernmethoden mit den oben beschriebenen Kernfunktionen signifikant bessere Ergebnisse erzielen als mit herkömmlichen Kernfunktionen (auf einem $2,5\%$ Signifikanzniveau). Gleichermaßen sind weg-basierte Kernfunktionen signifikant besser als zyklus-basierte Kernfunktionen, die aber auf diesem Datensatz, der nur wenige Moleküle mit vielen Zyklen enthält, wesentlich schneller berechnet werden können.

5 Zusammenfassung

Diese Arbeit erweiterte Kernmethoden auf allgemeinere, so genannte strukturierte, Daten. Dazu wurden passende Kernfunktionen definiert und charakterisiert. Empirische Ergebnisse in Anwendungen aus dem Bereich der pharmazeutischen Wirkstoffforschung zeigten substanzielle Verbesserungen gegenüber dem Einsatz konventioneller Lernverfahren.

Herkömmliche Kernmethoden können nun mit diesen Kernfunktionen dazu verwendet werden, aus einer vorgegebenen Datenbank von Molekülen solche auszuwählen, die viel versprechend sind und als nächstes im Labor getestet werden sollen. Somit ist eine Verringerung der Entwicklungszeit und -kosten für neue Medikamente möglich.

In Zukunft wollen wir nun Kernmethoden entwickeln, die in der Lage sind, ohne vorgegebene Datenbank von Molekülen sondern nur mit Hilfe einer kompakten Beschreibung der synthetisierbaren Moleküle, viel versprechende Moleküle vorzuschlagen. Diese können dann synthetisiert und im Labor getestet werden, in der Hoffnung auf neue wirksamere Medikamente mit weniger Nebenwirkungen.

Literatur

[Gär03] T. Gärtner. A Survey of Kernels for Structured Data. *SIGKDD Explorations*, 2003.

[Gär05] T. Gärtner. *Kernels for Structured Data*. PhD thesis, Universität Bonn, 2005.

[GFW03] T. Gärtner, P. A. Flach, and S. Wrobel. On Graph Kernels: Hardness Results and Efficient Alternatives. In *Proceedings of the 16th Annual Conference on Computational Learning Theory and the 7th Kernel Workshop*, 2003.

[GLF04] T. Gärtner, J. W. Lloyd, and P. A. Flach. Kernels and Distances for Structured Data. *Machine Learning*, 2004.

[Hau99] D. Haussler. Convolution Kernels on Discrete Structures. Technical report, Department of Computer Science, University of California at Santa Cruz, 1999.

[HGW04] T. Horvath, T. Gärtner, and S. Wrobel. Cyclic Pattern Kernels for Predictive Graph Mining. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 2004.

[IK00] W. Imrich and S. Klavžar. *Product Graphs: Structure and Recognition*. John Wiley, 2000.

[Llo03] J. W. Lloyd. *Logic for Learning*. Springer-Verlag, 2003.

[RT75] R. C. Read and R. E. Tarjan. Bounds on backtrack algorithms for listing cycles, paths, and spanning trees. *Networks*, 5(3), 1975.

[Tik63] A. N. Tikhonov. Solution of Incorrectly Formulated Problems and the Regularization Method. *Soviet Math. Dokl*, 1963.

[Wah90] G. Wahba. *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, 1990.

[Wat99] C. Watkins. Kernels from Matching Operations. Technical report, Department of Computer Science, Royal Holloway, University of London, 1999.

Thomas Gärtner hat 1999 an der Berufsakademie Mannheim in Informationstechnik diplomiert, 2000 einen “Master of Science” Kurs in “Advanced Computer Science” an der Universität Bristol absolviert und 2005 an der Universität Bonn promoviert. Seine Forschungsinteressen konzentrieren sich auf das Gebiet des Maschinellen Lernens, insbesondere auf die Teilgebiete Kernmethoden und Lernen mit strukturierten Daten. Er hat einige eingeladene Vorträge auf Workshops und Sommerschulen gehalten, hat ein Tutorial auf der internationalen Konferenz zu maschinellem Lernen (ICML) gehalten und hat einige Einladungen zu Forschungsaufenthalten und Gastvorträgen an renommierten Universitäten und Forschungsinstituten im In- und Ausland wahrgenommen. Er ist Autor oder Coautor von mehr als 20 wissenschaftlichen Publikationen, ist Mitglied in den Programmkomittees vieler internationaler Konferenzen, ist Vorsitzender für ein Gebiet bei der diesjährigen Europäischen Konferenz zu maschinellem Lernen (ECML) und ist Mitglied im Herausgebergremium der Zeitschrift “Machine Learning”.