

Abschlussbericht für das Karl-Steinbuch-Stipendium

Klassifikatorfusion auf Microarray-Daten

Christoph Müssel

15.05.2008

Zusammenfassung

Ziel meines Projekts war die Entwicklung und Evaluierung von Klassifikatoren-Ensembles zur Analyse von hochdimensionalen Microarray-Daten. Bei der Set Covering Machine handelt es sich um einen Klassifikator, der aus einer Menge von binären Merkmalen Konjunktionen und Disjunktionen lernt. Schon eine einzelne Set Covering Machine (SCM) stellt eine Fusion mehrerer einfacher Entscheidungen auf einzelnen Merkmalen dar. Aufgrund der hohen Dimension der Microarray-Daten findet der Lernalgorithmus hier mehrere gleichwertige Klassifikatoren, so dass eine Fusion dieser Lösungen möglich ist.

Im Verlauf des Projekts wurden Fusionsverfahren entwickelt und getestet, die mehrere SCMs fusionieren. Die Erkennungsrate der Verfahren wurde mit Hilfe von Kreuzvalidierungen auf verschiedenen Datensätzen evaluiert.

Danach folgte eine Analysephase, deren Ziel es war, die Charakteristika von Datensätzen zu ermitteln, bei denen eine solche Klassifikatorfusion zum Erfolg führt.

1 Verfahren

1.1 Die Set Covering Machine

Marchand und Shawe-Taylor[9] erweitern mit der Set Covering Machine die Algorithmen von Valiant[11] und Haussler[6], die aus einer Menge von binären Merkmalen eine kleinstmögliche Untermenge bilden, um eine Menge von Mustern abzudecken. Das zugrundeliegende Lernproblem lässt sich auf das NP-vollständige *Minimum Set Cover Problem* zurückführen. Bei den Algorithmen handelt es sich um Greedy-Heuristiken, die auf dem *Greedy Set Cover-Algorithmus* basieren. Dieser ist ein $(\ln n + 1)$ -Approximationsalgorithmus und liefert damit bereits eine recht gute Approximation [3].

Die Neuerung bei der SCM gegenüber den Algorithmen von Valiant und Haussler besteht in der Parametrisierbarkeit, über die es möglich ist, Fehler auf der Trainingsmenge zuzulassen und damit u. U. die Generalisierungsfähigkeit des Klassifikators zu verbessern. Je nach Konfiguration des Lernalgorithmus wird eine Konjunktion oder eine Disjunktion von Merkmalen gelernt. Dieser Klassifikator ist dann auch von Menschen gut interpretierbar. Durch die Beschränkung auf binäre Entscheidungen ist die SCM nur auf Zwei-Klassen-Probleme anwendbar.

1.2 Data-Dependent Rays

Microarray-Daten liegen üblicherweise als reellwertige Matrizen vor, so dass für den Einsatz der SCM erst eine Binarisierung der Merkmale vonnöten ist. Kestler et al.[7] schlagen die Aufteilung der Merkmale in zwei halboffene Intervalle mit Hilfe eines aus den Merkmalsausprägungen entnommenen Schwellwerts vor. Für diese Methode lässt sich eine obere Schranke des Generalisierungsfehlers der SCM berechnen, die zur Bewertung des Klassifikators genutzt werden kann.

In meinem Projekt wurden sämtliche möglichen Intervalle anhand der Merkmalsausprägungen berechnet und das Schwellwertverfahren auf den ursprünglichen Merkmalsraum angewandt. Der resultierende binäre Merkmalsraum diente als Eingabe für das Training der SCM.

Im Gegensatz zu den meisten gebräuchlichen Klassifikatoren (z. B. Support Vector Machines, Nearest Neighbour, Künstliche Neuronale Netze) ist die Set Covering Machine in Kombination mit Data-Dependent Rays nicht distanzbasiert. Es handelt sich hier um eine Fusion von Einzelentscheidungen, die wiederum auf Schwellwerten basieren. Folglich wird lediglich die Anordnung der Daten bezüglich dieses Schwellwertes berücksichtigt – es müssen keine weiteren Annahmen über die Eigenschaften der Daten gemacht werden.

1.3 Fusion

Der Trainings-Algorithmus der SCM erzeugt meist mehrere Klassifikatoren, da es in den Auswahl-schritten des Algorithmus oft mehrere Merkmale gibt, die die Klassen gleich gut trennen. Daraus resultiert ein Baum von Klassifikatoren, die teilweise Merkmale gemeinsam nutzen. Diese Einzelklassifikatoren können fusioniert werden – möglich ist auch eine Vorauswahl der besten Klassifikatoren mithilfe der theoretischen Schranke des Generalisierungsfehlers, wie in [7] beschrieben. Damit kann das Klassifikatorfusionsverfahren auch unabhängig von einer Sensorfusion verwendet werden – d. h. es ist auch möglich, das Verfahren anzuwenden, wenn für jeden Patienten nur eine Art von Befund (Sensor) verfügbar ist. Das ist bei den meisten verfügbaren Microarray-Datensätzen der Fall. Die Klassifikationen der einzelnen SCMs ergeben für jedes Muster einen binären Vektor. Aufgrund dieses Vektors kann nun durch ein Fusionsverfahren eine Gesamtentscheidung getroffen werden. Hierfür gibt es zwei Klassen von Verfahren:

1.3.1 Untrainierte Fusionsverfahren

Diese Verfahren treffen eine Entscheidung, ohne dabei durch Training an die Eingabedaten angepasst zu werden. Es wurden folgende Verfahren getestet:

- **Konjunktion:** Das Muster wird Klasse 1 zugeordnet, wenn alle Einzelklassifikatoren es dieser Klasse zuordnen würden. Andernfalls wird es Klasse 2 zugeordnet. Wurden die einzelnen SCMs als Disjunktion trainiert, erhält man eine Konjunktive Normalform (KNF), deren Literale Merkmale sind.
- **Disjunktion:** Das Muster wird Klasse 1 zugeordnet, wenn mindestens ein Einzelklassifikator es dieser Klasse zuordnen würde. Andernfalls wird es Klasse 2 zugeordnet. Wurden die einzelnen SCMs als Konjunktion trainiert, erhält man eine Disjunktive Normalform (DNF).
- **Majority Vote:** Das Muster wird der Klasse zugeordnet, die die meisten Stimmen der Einzelklassifikatoren enthalten hat. Hierfür muss die Anzahl der Klassifikatoren ungerade sein, da es sonst zu einem Gleichstand und damit zu einem undefinierten Ergebnis kommen kann.

1.3.2 Trainierte Fusionsverfahren

Für diese Verfahren wird eine separate Trainingsmenge benötigt, um sie an die Eingangsdaten anzupassen. Das Vorgehen für das Training eines solchen Verfahrens ist daher wie folgt:

1. Die Trainingsmenge wird in zwei Teile geteilt. Hier wurden gleiche Größen für beide Teilmengen gewählt.
2. Die SCMs werden mit der ersten Teilmenge trainiert.
3. Die Muster der zweiten Teilmenge werden mit Hilfe der trainierten SCMs klassifiziert.
4. Die binären Ausgabevektoren der SCMs für diese Muster werden als Trainingsmenge für das Fusionsverfahren verwendet.

Der Vorteil dieser Verfahren ist, dass sie sich an die Charakteristika der Einzelklassifikatoren anpassen und beispielsweise Schwächen einzelner Klassifikatoren auf bestimmten Unterräumen des Merkmalsraumes ausgleichen können. Nachteilig wirkt sich dagegen aus, dass die Einzelklassifikatoren mit einer halb so großen Trainingsmenge trainiert werden müssen. Dies kann eine geringere Anpassung der Einzelklassifikatoren an die Daten zur Folge haben.

Bisher wurden folgende Verfahren getestet:

- **Nearest Neighbour:** Ein Muster (binärer Ausgabevektor der SCMs) wird derjenigen Klasse zugeordnet, der das Trainingsmuster angehört, zu dem es den geringsten Abstand hat. Als Abstandsmaß wurde die Hamming-Distanz verwendet.
- **Nearest Prototype:** Aus den Trainingsmustern wird ein Mittelwert (Prototyp) für jede Klasse gebildet. Ein Muster wird derjenigen Klasse zugeordnet, deren Prototyp den geringsten Abstand dazu hat. Hier wurde als Abstandsmaß die Euklidische Distanz verwendet.

1.4 Testverfahren

1.4.1 n -fache Kreuzvalidierung

Dieses Testverfahren ist besonders geeignet, um bei einer geringen Anzahl von Mustern eine Aussage über die Erkennungsleistung eines Klassifikators zu machen. Ein mit Klassenlabels versehener Datensatz wird dabei in n zufällige Teilmengen unterteilt. Nun wird der Klassifikator n -mal jeweils mit den Mustern aus $n - 1$ Teilmengen trainiert, während die – wechselnde – n -te Teilmenge zum Test des Klassifikators verwendet wird. Die Klassifikationsfehler der n Durchläufe werden aufsummiert und geben im Verhältnis mit der Gesamtzahl der Muster einen prozentualen Fehler an. Um das Ergebnis zu stabilisieren, kann der beschriebene Vorgang mehrfach wiederholt werden. Für meine Tests verwendete ich eine 10×5 -fache Kreuzvalidierung.

2 Daten

2.1 Der Golub-Datensatz

Der Leukämie-Datensatz von Golub et al.[5] besteht ursprünglich aus 6817 Genen und 72 Mustern in 2 Klassen (47 ALL und 25 AML). Durch die Vorverarbeitung von Dudoit et al. [4] wird die Anzahl der Merkmale auf 3051 reduziert. Der Datensatz ist durch ein einziges Merkmal trennbar und daher eher einfach.

2.2 Der Khan-Datensatz

Der Zelltumor-Datensatz von Khan et al.[8] besteht aus 2308 Merkmalen und 63 Muster. Die Muster sind in 4 Klassen aufgeteilt (23 EWS, 20 RMS, 12 NB, 8 NHL). Um den Datensatz für die binäre SCM-Klassifikation einsetzen zu können, wurden aufgrund der hierarchischen Clusterung in [8] die 3 Klassen EWS, RMS und NB zu einer Klasse zusammengefasst.

2.3 Der Pomeroy-Datensatz

Dieser Datensatz von Pomeroy et al.[10] enthält Daten embryonaler Tumoren des zentralen Nervensystems. Es handelt sich hier um Datensatz C aus dem Paper. Er enthält 7129 Gene und 60 Muster, von denen 39 zu Klasse 1 (an Erkrankung gestorben) und 21 zu Klasse 2 (Überlebende) gehören.

2.4 Der Alon-Datensatz

Der Dickdarm-Tumor-Datensatz von Alon et al.[1] enthält in der verwendeten Version 2000 ausgewählte Gene und 62 Muster, von denen 40 aus Tumorgewebe (“negative”) und 22 aus Normalgewebe (“positive”) stammen.

2.5 Der Diagnostic Chip-Datensatz

Der Diagnostic Chip-Datensatz von Buchholz et al. enthält 62 Muster (37 Pankreas-Karzinome und 25 Pankreatitis bzw. Normalgewebe) mit 169 Merkmalen [2]. Die in diesem Datensatz verwendeten Gene sind dafür bekannt, mit Krebserkrankungen in Verbindung zu stehen.

3 Ergebnisse

In den Tests der ersten Phase wurde mit Hilfe von Kreuzvalidierungen untersucht, welche Fusionsverfahren auf den gegebenen Datensätzen am besten arbeiten und wie sich das fusionierte Verfahren im Vergleich zu einzelnen SCM-Klassifikatoren verhält. Das Training der SCMs wurde dabei so eingeschränkt, dass pro Schritt im Greedy-Algorithmus maximal 2 Verzweigungen erlaubt waren, da mit der Anzahl der Schritte die Anzahl der unterschiedlichen SCMs exponentiell wächst.

Die Tests deuten darauf hin, dass eine trainierte Fusion einer untrainierten Fusion überlegen ist. Bei einigen Datensätzen verbesserte die Fusion die Klassifikationsergebnisse erheblich, während sie bei anderen auch zu leicht schlechteren Ergebnissen führte. Um eine genauere Analyse dieses Verhaltens zu ermöglichen, wurden die Parameter der SCM in der zweiten Phase so eingestellt, dass sie sich wie der Haussler-Algorithmus verhält. Diese Vereinfachung stellt die Vergleichbarkeit der Ergebnisse sicher und macht das Verhalten des Trainingsalgorithmus leichter erklärbar. Sofern nicht anders vermerkt, handelt es sich bei den hier vorgestellten Ergebnissen um die Ergebnisse dieser Vereinfachung. Im Folgenden werden die Ergebnisse der Fusion mehrerer SCMs mit dem *Nearest Prototype*-Klassifikator genauer diskutiert.

Auf dem **Golub-Datensatz** findet der SCM-Lernalgorithmus tatsächlich ein einzelnes Merkmal, das die Trainingsmenge vollständig trennt. In der Fusion wird deutlich, dass es zwei gleichwertige Merkmale gibt, die die Klassen trennen, denn hier werden jeweils zwei SCMs mit einem Merkmal fusioniert. Dabei halbiert sich der mittlere Fehler durch die Fusion von $0,4/38$ (1,05%) auf $0,2/38$ (0,53%). Bemerkenswert ist, dass in 8 der 10 Kreuzvalidierungs-Läufe kein einziger Fehler auf der Testmenge gemacht wurde. Allerdings ist auch der mittlere Fehler der unfusionierten SCMs schon hervorragend.

Auch der **Khan-Datensatz** lässt sich bei der gegebenen Klassenaufteilung (von einem 4-Klassen-Problem in ein 2-Klassen-Problem) mit einem einzigen Merkmal trennen. Auch hier sind 8 der 10 Kreuzvalidierungs-Läufe komplett fehlerfrei. Durch die Fusion halbiert sich der Fehler hier von $0,65/63$ (1,03%) ohne Fusion auf $0,3/63$ (0,48%) mit Fusion.

Auf dem **Pomeroy-Datensatz** ist die Fehlerrate des fusionierten Klassifikators mit $5,9/60$ (9,83%) leicht schlechter als der durchschnittliche Fehler der einzelnen SCMs, der bei $4,69/60$ (7,82%) liegt. Eine Verschlechterung der Fusion gegenüber den unfusionierten Klassifikatoren lässt sich dadurch erklären, dass den SCMs innerhalb des fusionierten Klassifikators nur die Hälfte der Trainingsdaten zur Verfügung steht. Allerdings ist der fusionierte Klassifikator in den meisten Fällen immer noch deutlich besser als die jeweils schlechteste einzelne SCM. Möglicherweise lässt sich also das Risiko eines hohen Fehlers, das bei der Wahl eines bestimmten Einzelklassifikators besteht, durch eine Fusion verringern.

Dieser Datensatz ist deutlich schwieriger als die vorhergehenden, was sich auch an der höheren Anzahl der Merkmale erkennen lässt, die von den SCMs ausgewählt wurden. In den meisten Fällen wurden 4 Merkmale gewählt, diese Zahl stieg aber in Extremfällen bis auf 38.

Beim **Alon-Datensatz** hängt die Performance des Klassifikators auch von der Parameterwahl ab. In ersten Tests, bei denen der Penalty-Parameter der SCM auf einen festen Wert gesetzt war, zeigte die Fusion eine leichte Überlegenheit des fusionierten Klassifikators. Dieser erreichte einen Fehler von $3,6/62$ (5,81%), während die einzelnen SCMs einen leicht höheren durchschnittlichen Fehler von $4,53/62$ (7,3%) erreichte.

Die oben beschriebene Vereinfachung des Verfahrens zum Haussler-Algorithmus ging dagegen mit einer Verschlechterung der Ergebnisse einher, der nun mit einem Fehler von $4,7/52$ (7,58%) leicht schlechter war als die einzelnen SCMs $4,51/62$ (7,29%).

Der **Diagnostic Chip-Datensatz** gilt als sehr schwieriger Datensatz. Dies spiegelt sich auch in den Fehlerraten wider. Das fusionierte Verfahren erreicht einen Fehler von $5,2/62$ (8,39%), während die einzelnen SCMs mit einem durchschnittlichen Fehler von $4,55/62$ (7,35%) leicht besser abschneidet. Die Anzahl der gewählten Merkmale in den einzelnen Läufen der Kreuzvalidierung schwankt sehr stark – bei den einzelnen SCMs zwischen einem und 48 Features. Beim Fusionsverfahren summieren sich die Merkmale der einzelnen SCMs auf Werte zwischen 3 und 488 Merkmalen. Das deutet darauf hin, dass es der Set Covering Machine auf diesem Datensatz nicht gelingt, eine stabile Lösung zu finden, so dass das Hinzunehmen und Entfernen von Mustern in der Kreuzvalidierung zu drastisch unterschiedlichen Klassifikatoren führt.

Zur genaueren Analyse der Gründe dieses unterschiedlichen Verhaltens wurde der Haussler-Algorithmus jeweils nochmals auf dem gesamten Datensatz trainiert. Die entstandenen Klassifikatoren wurden untersucht und mit den zugehörigen Testergebnissen verglichen. Eine allgemeine Aussage scheint dabei zu sein, dass das Verbesserungspotential durch Fusionsverfahren von der Diversität des Datensatzes abhängt. Das bedeutet, dass eine Fusion mehrerer gleichartiger Klassifikatoren dann zu einer Verbesserung gegenüber einzelner Klassifikatoren führt, wenn die fusionierten Klassifikatoren einen möglichst unterschiedlichen Bereich des Datensatzes abdecken. Dadurch können Fehler einzelner Klassifikatoren im Ensemble durch die richtigen Ergebnisse anderer Klassifikatoren ausgeglichen werden. Überlappen sich dagegen die von den einzelnen Klassifikatoren des Ensembles sehr stark, machen sie auch dieselben Fehler und sind daher weniger robust. Dieses Verhalten kann in den Datensätzen sehr gut beobachtet werden:

Im Golub-Datensatz finden sich zwei Merkmale, die unabhängig voneinander den Datensatz kom-

plett trennen. Ähnliches gilt für den Khan-Datensatz, wo es sogar sechs solche Merkmale gibt. In diesen Fällen liegen also völlig voneinander unabhängige Klassifikatoren vor. Die obigen Kreuzvalidierungs-Ergebnisse bestätigen für diese Datensätze eine deutliche Verbesserung der Klassifikationsergebnisse durch die Fusion.

Beim Pomeroy-Datensatz dagegen findet der Trainingsalgorithmus überhaupt nur einen einzigen Klassifikator, der den Datensatz komplett trennt. Folglich ist es klar, dass eine Klassifikatorfusion hier keine Vorteile erbringen kann. Zwar sorgt die Reduktion der Trainingsmuster in der Kreuzvalidierung wieder dafür, dass sich in den einzelnen Läufen mehrere Lösungen finden, diese machen dann aber Fehler auf den Testdaten. Da sich die Trainingsmenge der fusionierten SCMs im Vergleich zu den einzelnen SCMs nochmals halbiert, verstärkt sich dieser Effekt, und die Fusion schneidet schlechter ab als die Einzelklassifikation.

Beim besonders schwierigen Diagnostic Chip-Datensatz kann ein ähnliches Verhalten beobachtet werden. Zwar gibt es hier eine Vielzahl von Lösungen, die den Datensatz komplett trennen. Jedoch sind alle diese Lösungen in den ersten 3 Merkmalen – die aufgrund der Greedy-Strategie des Algorithmus bei der Klassifikation das größte Gewicht haben – identisch. Auch hier ist es folglich nicht möglich, Lösungen zu finden, die stark unterschiedliche Bereiche des Merkmalsraums abdecken. Wie oben bereits beschrieben, führt das in diesem Fall ebenfalls zu leicht schlechteren Ergebnissen des Fusionsverfahrens.

4 Zusammenfassung

Die Entwicklung eines fusionsbasierten Klassifikators auf Grundlage der Set Covering Machine zeigt auf bestimmten Datensätzen bereits vielversprechende Ergebnisse bis hin zur Halbierung des Klassifikationsfehlers. In anderen Bereichen sind Einzelklassifikatoren jedoch weiterhin überlegen. Das eröffnet interessante Möglichkeiten für theoretische Betrachtungen bezüglich des Optimierungspotentials durch Klassifikatorfusion. In eingeschränktem Umfang wurden bereits – wie oben beschrieben – Untersuchungen zu den Ursachen der unterschiedlichen Klassifikationsleistungen durchgeführt. Jedoch sind in diesem Bereich noch weitere interessante Untersuchungen möglich, beispielsweise mithilfe künstlicher Datensätze. So verbleiben auch über das Projektende hinaus noch Ansatzpunkte zur weiteren Arbeit. Das Manuskript für eine wissenschaftliche Publikation über die Fusion einfacher Entscheidungsregeln ist bereits in Vorbereitung. Dies ist ein direktes Ergebnis des Projekts und steht in Kontinuität zu vorherigen Arbeiten.

Literatur

- [1] ALON, U., N. BARKAI, D. A. NOTTERMAN, K. GISH, S. YBARRA, D. MACK und A. J. LEVINE: *Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays*. Proceedings of the National Academy of Sciences, 96(12):6745–6750, 1999.
- [2] BUCHHOLZ, M., H. A. KESTLER, A. BAUER, W. BOCK, B. RAU, G. LEDER, W. KRATZER, M. BOMMER, A. SCARPA, M. SCHILLING, G. ADLER, J. HOHEISEL und T. GRESS: *Specialized DNA arrays for the differentiation of pancreatic tumors*. Clin. Cancer Res., 11(22):8048–54, 2005.
- [3] CORMEN, T. H., C. E. LEISERSON, R. L. RIVEST und C. STEIN: *Introduction to Algorithms, Second Edition*. The MIT Press, September 2001.
- [4] DUDOIT, S., J. FRIDLYAND und T. SPEED: *Comparison of discrimination methods for the classification of tumors using gene expression data*. Journal of the American Statistical Association, 97(457):77–87, 2002.
- [5] GOLUB, T., D. SLONIM, P. TAMAYO, C. HUARD, M. GAASENBEEK, J. MESIROV, H. COLLIER, M. LOH, J. DOWNING, M. CALIGIURI, C. BLOOMFIELD und E. LANDER: *Molecular classification of cancer: class discovery and class prediction by gene expression monitoring*. Science, 286:531–536, 1999.
- [6] HAUSSLER, D.: *Quantifying Inductive Bias: AI Learning Algorithms and Valiant’s Learning Framework*. Artif. Intell., 36(2):177–221, 1988.
- [7] KESTLER, H. A., W. LINDNER und A. MÜLLER: *Learning and Feature Selection Using the Set Covering Machine with Data-Dependent Rays on Gene Expression Profiles*. In: ANNPR, Seiten 286–297, 2006.
- [8] KHAN, J., J. WEI, M. RINGNER, L. SAAL, M. LADANYI, F. WESTERMANN, F. BERTHOLD, M. SCHWAB, C. ANTONESCU, C. PETERSON und P. MELTZER: *Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks*. Nature Medicine, 7(6):673–679, 2001.
- [9] MARCHAND, M. und J. SHAWE-TAYLOR: *The Set Covering Machine*. Journal of Machine Learning Research, (3):723–746, 2002.
- [10] POMEROY, S. L., P. TAMAYO, M. GAASENBEEK, L. M. STURLA, M. ANGELO, M. E. McLAUGHLIN, J. Y. H. KIM, L. C. GOUMNEROVA, P. M. BLACK, C. LAU, J. C. ALLEN, D. ZAGZAG, J. M. OLSON, T. CURRAN, C. WETMORE, J. A. BIEGEL, T. POGGIO, S. MUKHERJEE, R. RIFKIN, A. CALIFANO, G. STOLOVITZKY, D. N. LOUIS, J. P. MESIROV, E. S. LANDER und T. R. GOLUB: *Prediction of central nervous system embryonal tumour outcome based on gene expression*. Nature, 415(6870):436–42, 2002.
- [11] VALIANT, L. G.: *A theory of the learnable*. In: STOC ’84: Proceedings of the sixteenth annual ACM symposium on Theory of computing, Seiten 436–445, New York, NY, USA, 1984. ACM.