Prototypen basiertes maschinelles Lernen in der klinischen Proteomik

Frank-Michael Schleif

Universität Leipzig
AG Computational Intelligence,
Karl-Tauchnitz-Str. 25, 04107 Leipzig
schleif@informatik.uni-leipzig.de

Abstract: Die klinische Proteomik befasst sich mit der Untersuchung von Krankheitsbildern auf Basis von Proteinanalysen. Die dabei am häufigsten verwendete Messmethode ist die Massenspektrometrie. Dabei entstehen hochdimensionale Spektren, die eine problemangepasste Vorverarbeitung sowie Algorithmik für die Erzeugung von statistischen Modellen erfordern. Im Bereich klinischer Fragestellungen sollten die dabei eingesetzten Werkzeuge leicht interpretierbar sein, um ein tieferes Verständnis der klinischen Daten oder Anwendungen, wie zum Beispiel der Suche nach Krankheitsmarkern, zu gestatten. Prototypen basierte Algorithmen erweisen sich dabei als besonders günstig. In diesem Beitrag werden wesentliche Erweiterungen von prototypen basierten Verfahren skizziert, die den besonderen Herausforderungen der klinischen Proteomik Rechnung tragen. Die Verfahren werden um Metrikadaptation zur besseren Approximation der Klassengrenzen, Fuzzy-Klassifikation zur Modellierung unscharfer Klassen, sowie Konzepte des aktiven Lernens zur trennbreiten basierten Optimierung der Modelle erweitert und auf klinischen Datensätzen getestet.

1 Einführung

Die klinische Proteomik untersucht Krankheitsprozesse, deren Ausprägungen durch die Analyse des Proteinhaushalts der Individuen identifiziert werden können. Das Proteom, die Gesamtheit aller Proteine eines Organismus, ist dabei ein potentieller Indikator für eine Erkrankung. Proteine sind komplexe Substanzen und benötigen aufgrund ihrer hohen Variabilität eine besonders standardisierte Prozesskette für eine Analyse. Die Messung der Probe erfolgt dabei typischer Weise durch ein Massenspektrometer (siehe Abb. 1). Dabei entstehen hochdimensionale Spektren, die je nach Art der biochemischen Vorbereitung (siehe Abb. 1) unterschiedliche Aspekte der Probe charakterisieren. Allgemein können durch die Analyse der Spektren Masse-zu-Ladungsverhältnisse (MLV) von Ionen bestimmt werden. Für den Fall der klinischen Proteomik betrifft dies das Vorhandensein bzw. die Expressivität von bestimmten Proteinfragmenten. Eine weitere Herausforderung in der Analyse ist die eher geringe Anzahl von Proben im Verleich zur hohen Anzahl von spektralen Merkmalen. Neben der Komplexität der Proben spielt auch der klinische Aspekt eine wesentliche Rolle. So ist die Güte der Klassifikationsentscheidung von besonderer Bedeutung, die Interpretierbarkeit der Modelle, sowie deren Adaptierbarkeit bei Nachmessungen. Um diesen Aspekten Rechnung zu tragen werden die Spektren zunächst geeignet vorbereitet. Dabei wird eine reduzierte Darstellung der Spektren erstellt, die alle relevanten Informationen enthält. Für die so reduzierten Spektren können zum Beispiel

Klassifikationsmodelle erstellt werden. Nach geeigneter Evaluierung, können diese für die Analyse und Diagnostik von Krankheitsprozessen in Frage kommen. In diesem Artikel betrachten wir zunächst die Vorbereitung der Spektren, nachfolgend werden Konzepte prototypischer Klassifikationsverfahren beschrieben und deren Erweiterungen für die klinische Proteomik skizziert. Im Ergebnisteil wird die entwickelte Algorithmik zur Bildung von Klassifikationsmodellen für verschiedenen klinische Datensätze eingesetzt und zusammenfassend bewertet.

2 Massenspektrometrie in der klinischen Proteomik

Die Massenspektrometrie ist eine analytische Technik zur Messung von MLVs von Ionen in chemischen Strukturen. Dies wird durch Ionisation und Aufspaltung der Probe für verschiedenen Massen erreicht (siehe auch [Lie02]). Ein typisches Massenspektrometer is in Abbildung 1 schematisch dargestellt. Das Prinzip basiert auf dem Fakt, dass verschiedene

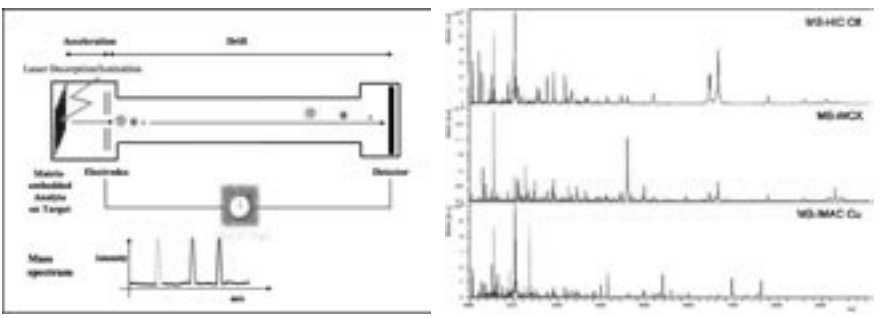

Abbildung 1: Schematische Abbildung wesentlicher Teile eines MALDI-TOF Massenspektrometers. Links wird die in Matrix eingebettete Probe auf den Probenträger aufgetragen und danach ionisiert. Dabei wird das Material vom Probenträger in Richtung Detektor beschleunigt, wobei die Probe weiter fragmentiert. Unterschiedliche molekulare Massen der Probe benötigen, bei gleicher zugeführter Energie, unterschiedlich lange, um die Flugstrecke zurückzulegen. Aus diesen Informationen kann die Struktur der Probe ermittelt werden. Die rechte Abbildung zeigt unterschiedliche Spektren ein und derselben Probe in Abhängigkeit der biochemischen Vorbereitung. Auf der x-Achse sind die Massenpositionen ($1 - 10$kDa) und auf der y-Achse die entsprechenden Intensitäten abgebildet.

chemische Strukturen unterschiedliche atomare Massen besitzen. Die Probe wird ionisiert und die Ionen werden in einer Beschleunigerkammer mit einem Magnetfeld senkrecht zur Richtung der Feldlinien beschleunigt, auf eine Kurvenbahn abgelenkt und zum Detektor geführt. Leichtere Ionen werden stärker abgelengt und erreichen den Detektor entsprechend früher. Der Detektor misst wie stark jedes Ion abgelengt wurde und ermittelt daraus das MLV. Auf Basis dieser Information kann die chemische Zusammensetzung der Probe bis zu einem gewissen Grad bestimmt werden. Die gemessenen Spektren bestehen aus mehreren 10000 Messpunkten, wobei lediglich Bereiche lokal höherer Intensität (Peaks) von Bedeutung sind, da diese auf das vorhandensein bestimmter Fragemente in der Probe hinweisen. Für eine adequate Vergleichbarkeit der Spektren werden diese zunächst basislinienkorrigiert (charakteristisches Artefakt des Messprozesses) [SS04, WNP+04, PKH06] und aligniert, z.B. über einen Least-Squares-Fit, so dass Peaks an korrespondierende Massenpositionen korrekt verglichen werden können (Details siehe [Sch06]). Die Identifika-

tion der Peaks (Peakpicking) geschieht dabei über eine lokale Maxima Suche die unter anderem auch Eigenschaften wie Massenposition, minimale Peakbereite und die Position von Umgebungspeaks mit berücksichtigt. Nachfolgend wird die Liste von Peakkandidaten anhand von Signal zu Rausch (S/N) Abschätzungen und Anforderungen an die minimale Intensität weiter reduziert. Am Ende der Datenvorverabeitung ist damit jedes Spektrum als eine sogenanntes Linienspektrum repräsentiert, welches nur noch aus verschiedenen Peaks besteht. Dabei nutzen alle Spektren ein identisches Gitter von Peakpositionen. Die vorhandene Liste von Peaks pro Spektrum wird genutzt um bestimmte Merkmale für weitere Analysen abzuleiten, hier Peakflächen. Die dadurch erhaltene Matrix von Merkmalen dient als Eingaberaum für die Generierung von Klassifikationsmodellen. Wesentlich bei dieser Art der Vorgehensweise ist, dass zu einem Merkmal in der Matrix stets die relative Massenposition im Originalspektrum erhalten bleibt. Dies ist relevant um, bei der Analyse von Gruppenunterschieden die dem diskrimierenden Merkmal zugrundeliegende Masse weiteren Messungen unterziehen zu können (ms/ms-Analyse). Diese Forderung begründet auch, warum bestimmte komplexere Modellmethoden nicht oder nur mit Mühe anwendbar sind (zum Beispiel nichtlineare Kernelmodelle), da dabei meist verschiedene oder alle Merkmale nichtlinear verrechnet werden und eine Massenidentifikation sehr erschwert ist.

3 Datenrepräsentation mit Prototypen

Klassifizierungs- und Clusteralgorithmen sind wesentliche Methoden bei der Mustererkennung, die in vielen Problemstellungen Anwendung finden. Prototypenbasierte Vektorquantisierungsmethoden haben sich dabei als robuste adaptive Modelle etabliert. Gewöhnliche Vektorquantisierung ist eine prototypenbasierte Klassifikationstechnik, die hauptsächlich durch die Standardverfahren LVQ1...LVQ3 [Koh95], beeinflusst ist. Eine Vielzahl von Erweiterungen verbessert die ursprünglichen Ansätze unter theoretischen und praktischen Aspekten [HV03, SBO03]. Die erste Variante, die eine Kostenfunktion optimiert, ist der sogenannte Generalized LVQ (GLVQ) [SY95] Algorithmus mit den entsprechenden Erweiterungen - Supervised Neural GAS (SNG) und Supervised Relevance Neural GAS (SRNG) [HSV05b]. Dabei sind Eingaben durch $\mathbf{v}$ mit Klassenbezeichner $c_{\mathbf{v}} \in \mathcal{L}$ gegeben. Sei $\mathcal{L}$ die Menge der Klassenbezeichner mit $\#\mathcal{L} = N_{\mathcal{L}}$ und $V \subseteq \mathbb{R}^{D_V}$ eine endliche Eingabemenge. LVQ verwendet eine fixe Anzahl von Prototypen (auch Gewichtsvektoren) für jede Klasse. Sei $\mathbf{W} = \{\mathbf{w_r}\}$ die Menge von Prototypen und $c_{\mathbf{r}}$ die Klassenbezeichner von $\mathbf{w_r}$. Desweiteren, sei $\mathbf{W}_c = \{\mathbf{w_r} | c_{\mathbf{r}} = c\}$ die Untermenge der Prototypen, der Klasse $c \in \mathcal{L}$. Die Klassifikation wird dann durch die Abbildung Ψ in Form einer winner-take-all Regel realisiert, z.B. ein Eingabevektor $\mathbf{v} \in V$ wird auf den Prototypen abgebildet $\mathbf{s} \in A$ als Vektor $\mathbf{w}_s$ zu dem er am nächsten liegt $\mathbf{v}$,

$$\Psi_{\mathcal{V}\to\mathcal{A}} : \mathbf{v} \mapsto \mathbf{s}(\mathbf{v}) = \underset{\mathbf{r}\in A}{\operatorname{argmin}}\, d(\mathbf{v}, \mathbf{w_r}) \tag{1}$$

mit $d(\mathbf{v}, \mathbf{w})$ ein beliebiges Ähnlichkeitsmaß, im Allgemeinen als quadratische euklidische Metric. Der Prototyp $\mathbf{s}$ wird als Sieger bezeichnet. Die Untermenge

$$\Omega_{\mathbf{r}} = \{\mathbf{v} \in V : \mathbf{r} = \Psi_{V\to A}(\mathbf{v})\}$$

die auf einen bestimmten Prototypen abgebildet wird (entsprechend (1)), bildet das rezeptive Feld des Prototypen. Standard LVQ-Training adaptiert die Prototypen in der Art,

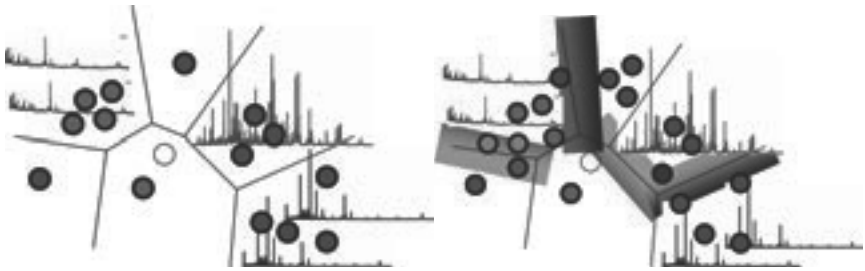

Abbildung 2: Schematische Abbildung prototypen basierter Klassifikation für 2d Daten mit multiplen Clustern. Die Prototypen sind jeweils durch einen zusätzlichen Kreis dargestellt und führen zu einer Zerlegung des Datenraums anhand der rezeptiven Felder (graue Netzstruktur). Die linke Abbildung betrachtet den Fall für fixe Klassenzuordnung, rechts entsprechend für unscharfe Klassenmodellierung. Unscharfe Übergangsbereiche, unter Berücksichtigung der Punktabstände sind schraffiert dargestellt.

dass für jede Klasse $c \in \mathcal{L}$, die entsprechenden Prototypen $\mathbf{W}_c$ die Klasse so genau wie möglich repräsentieren, z.B. sollten die Punkte einer gegebenen Klasse $V_c = \{\mathbf{v} \in V | c_\mathbf{v} = c\}$, und die Vereinigungsmenge $\mathcal{U}_c = \bigcup_{\mathbf{r}|_{\mathbf{w}_\mathbf{r} \in \mathbf{W}_c}} \Omega_\mathbf{r}$ der rezeptiven Felder der entsprechenden Prototypen sich sowenig wie möglich unterscheiden. Die Adaptation der Prototypen erfolgt entweder gemäß einer zu minimierenden Kostenfunktion als (stochastischer) Gradientenabstieg oder nach Heuristiken. Ziel ist somit die Generierung eines Modells welches die Daten anhand von Prototypen sowohl in ihrer Datenverteilung (unüberwacht) als auch in ihrer Klassenstruktur (überwacht) repräsentiert. Dabei können je nach Fragestellung und Verfahren die Prototyppositionen, die Klassenbezeichner und die verwendete Metrik optimiert werden. Nachfolgend geben wir eine kurze Einführung in ein typisches überwachtes prototypen basiertes Verfahren welches das Konzept der Nachbarschaftskooperation nutzt und eine Kostenfunktion optimiert, wir werden auf dieses Verfahren im folgenden mehrfach zurückkommen. Wesentlicher Aspekt ist dabei die Nachbarschaftskooperation beim Lernen, d.h. die Kooperativität der Prototypen untereinander beim Adaptationsprozess. Dies führt zu einer deutlichen Konvergenzverbesserung und erhöhten Robustheit der Modelle.

Soft Nearest Prototype Classification (SNPC) ist ein weiteres Prototypen basiertes Verfahren und wurde in [SBO03] eingeführt. Dabei werden sogenannte soft assignments für Datenvektoren zu den Prototypen definiert, die eine statistische Interpretation als normalisierte Gaussverteilungen besitzen. Im original SNPC betrachtet man:

$$E(\mathcal{S}) = \frac{1}{N_\mathcal{S}} \sum_{k=1}^{N_\mathcal{S}} \sum_{\mathbf{r}} u_\tau(\mathbf{r}|\mathbf{v}_k) \left(1 - \alpha_{\mathbf{r},c_{\mathbf{v}_k}}\right) \tag{2}$$

als eine Kostenfunktion mit $\mathcal{S} = \{(\mathbf{v}, c_\mathbf{v})\}$ als die Menge aller Eingabedaten, $N_\mathcal{S} = \#\mathcal{S}$. Die Klassenzuweisungsvariablen $\alpha_{\mathbf{r},c_{\mathbf{v}_k}}$ sind 1 wenn $c_{\mathbf{v}_k} = c_\mathbf{r}$ und 0 sonst. $u_\tau(\mathbf{r}|\mathbf{v}_k)$ ist die Wahrscheinlichkeit, dass ein Eingabevektor $\mathbf{v}_k$ dem Prototypen $\mathbf{r}$ zugewiesen wird. Eine crispe *winner-takes-all* Abbildung (1) ergibt $u_\tau(\mathbf{r}|\mathbf{v}_k) = \delta(\mathbf{r} = \mathbf{s}(\mathbf{v}_k))$.

Für die Minimierung von (2), in [SBO03] werden die Variablen $u_\tau(\mathbf{r}|\mathbf{v}_k)$ als soft assignment Wahrscheinlichkeiten behandelt. Dies gestattet einen Gradientenabstieg auf einer

Kostenfunktion (2) mit:

$$u_\tau\left(\mathbf{r}|\mathbf{v}_k\right) = \frac{\exp\left(-\frac{d(\mathbf{v}_k,\mathbf{w}_\mathbf{r})}{2\tau^2}\right)}{\sum_{\mathbf{r}'}\exp\left(-\frac{d(\mathbf{v}_k,\mathbf{w}_{\mathbf{r}'})}{2\tau^2}\right)} \tag{3}$$

und d als dem Abstandsmaß aus (1) und τ ist die Bandbreite der Gaussfunktion. Die Kostenfunktion (2) lässt sich wie folgt auch mit lokalen Kosten $lc\left(\left(\mathbf{v}_k, c_{\mathbf{v}_k}\right)\right)$ definieren:

$$E\left(\mathcal{S}\right) = \frac{1}{N_\mathcal{S}}\sum_{k=1}^{N_\mathcal{S}} lc\left(\left(\mathbf{v}_k, c_{\mathbf{v}_k}\right)\right) \quad lc\left(\left(\mathbf{v}_k, c_{\mathbf{v}_k}\right)\right) = \sum_{\mathbf{r}} u_\tau\left(\mathbf{r}|\mathbf{v}_k\right)\left(1-\alpha_{\mathbf{r},c_{\mathbf{v}_k}}\right) \tag{4}$$

Der lokale Fehler ist die Summe der Klassenzuweisungsvariablen $\alpha_{\mathbf{r},c_{\mathbf{v}_k}}$ zu allen Prototypen einer falschen Klasse, und, somit $lc\left(\left(\mathbf{v}_k, c_{\mathbf{v}_k}\right)\right) \leq 1$ mit lokalen Kosten, abhängig von der ganzen Menge $\mathbf{W}$. Aufgrund der lokalen Kosten $lc\left(\left(\mathbf{v}_k, c_{\mathbf{v}_k}\right)\right)$ (kontinuierlich und beschänkt), kann die Kostenfunktion (4) durch einen stochastischen Gradientenabstieg minimiert werden (Details in [SBO03]).

3.1 Relevance learning für SNPC

Wie alle Nächster-Prototype (NPC) Klassifikationsverfahren, hängt SNPC erheblich von der verwendeten Metrik ab. Für hochdimensionale Daten, wie sie zum Beispiel in der klinischen Proteomik anzutreffen sind, ist die Wahl der normalen euklidischen Metrik häufig ungünstig, da viele Eingabedimensionen, aus Sicht der Klassifikationsfragestellung, Rauschen kodieren. Diesem Fakt Rechnung tragend wurde der SNPC um die Methode des Relevanzlernen [HV02] erweitert, um die Eingabedimensionen zu bestimmen, die für den Klassifikationsprozess von Bedeutung sind.

Relevanzlernen, bietet die Möglichkeit, die Metrikparameter mit zu lernen. Dieses Konzept wird nun in SNPC eingeführt und nachfolgend als SNPC-R bezeichnet: Ein Parametervektor $\lambda = (\lambda_1, \ldots, \lambda_m)$ wird der verwendeten Metrik zugewiesen $d\left(\mathbf{v}_k, \mathbf{w}_\mathbf{r}\right)$ bezeichnet als $d^\lambda\left(\mathbf{v}_k, \mathbf{w}_\mathbf{r}\right)$, und daraufhin in den soft assignments (3) verwendet. Ein häufiges Beispiel ist die skalierte euklidische Metrik $d^\lambda\left(\mathbf{v}_k, \mathbf{w}_\mathbf{r}\right) = \sum_{i=1}^{D_V}\lambda_i(\mathbf{v}_k^i - \mathbf{w}_\mathbf{r}^i)^2$ Parallel zur Optimierung der Prototypen werden nun auch die Relevanzparameter λ_j entsprechend des Klassifikationsproblems, adaptiert. Dies geschieht durch einen Gradientenabstieg auf der Kostenfunktion entsprechend der Parameter (siehe [Sch06]) gefolgt von einer Normalisierung der λ_j. Die Metrikparameter können auch individuell für jeden Prototyp oder pro Klasse adaptiert werden, damit erhält man den lokalen SNPC-R Algorithmus (siehe [Sch06]). Im Falle der skalierten euklidischen Metrik können die λ_j in einem Relevanzprofil analysiert werden, welches die *Relevanz* jedes Merkmals bzw. Peaks für die Klassenseparation anzeigt. Auf Basis dieser Profile kann dann ähnlich zur Recursiven Feature Eliminierung [HTF01] ein Pruning der Eingabedimensionen oder in unserem Fall eine Reduktion der Linienspektren erfolgen.

In [CGBAA02] wurde gezeigt, dass man NPC, basierend auf euklidischer Metrik, als large margin Algorithmen mit dimensionsunabhängigen Generalisierungsschranken interpretieren kann, wobei also die Abstände zwischen den Trennebenen maximiert werden. Anstelle der Datendimension dient der Hypothesen-Abstand als Parameter der Generalisierungsschranke, welches der Abstand ist, der gerade noch geändert werden kann ohne die

Klassifikationsentscheidung zu verändern. In [HSV05b] konnte dieses Schemata für NPC-Algorithmen mit adaptiver Diagonalmetrik erweitert werden. In [Sch06] konnte gezeigt werden dass auch für lokale Metrikparameter ähnliche Generalisierungseigenschaften ableitbar sind, wobei diese lokalen Modelle neben einer grösseren Flexibilität der Modellierung zusätzlich lokal relevante Massenpositionen identifizierbar machen.

4 Fuzzy Klassifikation für SNPC-R

Für den *Fuzzy gelabelten* SNPC (FSNPC) werden nun fuzzy Werte für $\alpha_{\mathbf{r},c}$ zugelassen, die anzeigen, in wie weit ein Datenpunkt einem rezeptiven Feld zugewiesen ist so dass $0 \leq \alpha_{\mathbf{r},c} \leq 1$ im Gegensatz zum Crisp-Fall und unter der Normalisierungsbedingung $\sum_{c=1}^{N_{\mathcal{L}}} \alpha_{\mathbf{r},c} = 1$. Die Prototypklassenbezeichner werden automatisch während des Trainings adaptiert. Dabei zeigt sich allerdings, dass die klare Klassenformation, wie sie für die Lerndynamik des Standard SNPC vorausgesetzt wird, nicht länger verfügbar ist. Allerdings kann eine ähnliche Lerndynamik hergeleitet werden

$$\triangle \mathbf{w_r} = -\frac{T}{2\tau^2} \cdot \frac{\partial d_{\mathbf{r}}}{\partial \mathbf{w_r}} \quad T = u_\tau\left(\mathbf{r}|\mathbf{v}_k\right) \cdot \left(1 - \alpha_{\mathbf{r},c_{\mathbf{v}_k}} - lc\left(\mathbf{v}_k, c_{\mathbf{v}_k}\right)\right) \tag{5}$$

Parallel können die fuzzy labels $\alpha_{\mathbf{r},c_{\mathbf{v}_k}}$ optimiert werden $\triangle \alpha_{\mathbf{r},c_{\mathbf{v}_k}} = -u_\tau\left(\mathbf{r}|\mathbf{v}_k\right)$ gefolgt von einer Normalisierung. Im SNPC wird eine sogenannte Fensterregel verwendet, um den Algorithmus stabil zu halten. Wie in [Sch06] gezeigt wurde kann auch für den SNPC mit fuzzy Labeln eine ähnliche Fensterregel abgeleitet werden. Auch hier kann Relevanzlernen integriert werden [VSH06a] bezeichnet als FSNPC-R. Durch die Verwendung von Fuzzy-Labeln können die Prototypen selbst anhand der Datenverteilung einen Klassenbezeichner lernen und zudem werden schlecht gelernte Rezeptive Felder durch unsichere Klassenentscheidungen offenbar. Beide Aspekte sind hilfreich im Kontext der klinischen Proteomik, da einerseits die Anzahl der Prototypen pro Klasse nicht vorgegeben werden muss und zum anderen Zuordnungen von Datenpunkten zu überlappenden Bereichen identifiziert werden können.

5 Aktive Lernstrategien

Die bereits vorher erwähnte Margin Analyse aus [CGBAA02], [HSV05a] und [Sch06] motiviert elegante Schemata für aktive Lernstrategien. Offensichtlich, hängt die Generalisierungsfähigkeit von GRLVQ-Algorithmen lediglich von Punkten mit zu kleinem Margin ab. Somit müssen lediglich, extreme Marginwerte beschränkt werden und eine Einschränkung entsprechender Updates sollte nur für extreme Paare von Prototypen erfolgen. Diese Argumentation führt zur Entwicklung von aktiven Lernschemata wenn eine gegebene statische Menge von Trainingsdaten vorliegt. Dazu definiert man eine monoton abfallende, nichtnegative Selektionsfunktion (0 - keine Selektion, 1 - garantierte Selektion) $L^c : \mathbb{R} \rightarrow \mathbb{R}$ und selektiert aktiv Trainingsdatenpunkte von einer gegebenen Beispieldatenmenge (annealed mit α) als:

1. $L^c(t) = 1$ for $t < 0$ and $L^c(t) \sim |t|^\alpha$, sonst; (Probabilistische Strategie).

2. $L^c(t) = 1$ if $t \leq \rho$, sonst, 0; (Threshold Strategie).

Beide Strategien, zielen auf schlecht representierte Trainingspunkte und damit direkt auf eine Verbesserung der Generalisierungsschranken. Strategie (2) gestattet die Adaptation der Margin parameter ρ während des Trainings entsprechend der Konfidenz des Modells. Für jeden Prototypen $\mathbf{w_r} \in W$ wird dazu ein neuer Parameter α_r eingeführt, der die Distanz von Datenpunkten zum aktuellen Prototypen innerhalb des rezeptiven Felds misst. Wir wählen ρ_r lokal als $\rho_r = 2 \cdot \alpha_r$. Damit werden Punkte, deren Margin bereit groß im Verleich zum rezeptiven Feld ist, mit hinreichender Sicherheit repräsentiert und im Lernen ignoriert.

6 Anwendungen in der klinischen Proteomik

Das Ziel der Massenspektrometrie in der klinischen Proteomik ist die Generierung von Protein-Profilspektren, meist aus Körperflüssigkeiten wie Serum oder Urin. Biomarkermuster sind dabei versteckte komplexe Signale. Die verbesserten LVQ-Ansätze sind besonders geeignet, um eindeutige biomarker Muster aufzufinden. Individuelle Biomarker sind häufig unzureichend für die Krebsdiagnostik und die Aufdenkung von Biomarkermustern könnte eine bessere Diagnose ermöglichen auch bevor erste Sympthome auftreten. Dabei sollte der ideale Biomarker hohe Reproduzierbarkeit, Sensitivität, Spezifizität besitzen und auch als Indikator für verschiedene Stadien der Erkrankung geeignet sein, so dass auch ein Behandlungseffekt im Indikator sichtbar wird. Die meisten Erkrankungen sind durch unterschiedliche, überlappende Stadien gekennzeichnet, die zu unscharfen Klassifikationen führen. Daher werden lokale Klassifikationsmodelle als geeigneter angesehen, um mit klinischen Daten umzugehen. Nachfolgend werden die vorgestellten Methoden in der Analyse von klinischen Proteomdaten (Proteom I, Proteom II, WDBC aus [Sch06],[BM06]) angewandt und mit Standardverfahren (Support Vector Machine (SVM), Diskriminanzanalyse (LDA /QDA) sowie GLVQ und SNG) verglichen (Details in [Sch06],[HTF01]).

6.1 Klassifikation von Proteomspektren

Beginnende mit einfachen Ansätzen wurden für den Proteom II Datensatz Vorhersagegenauigkeiten von 81% mit LDA und 71% bei Verwendung von QDA auf Basis der ersten 30 (niedrige Massen) Merkmale errzielt. Für den Proteom I Datensatz wurde LDA/QDA lediglich auf den ersten 30ig Merkmalen berechnet, da sonst numerische Instabilitäten auftraten. Für diesen Datensatz konnten Ergebnisse von 81% und für QDA von 75% erzielt werden. Mit SVM und einem Polynomial Kernel, lassen sich 82% Vorhersagegenauigkeit für den Datensatz Proteom I und 89% für Proteom II ermitteln. Weiterhin wurden alternative Metriken wie z.B. die Tanimoto-Distanz (TM), die vorallem in der Chemie zur Analyse von Strukturähnlichkeiten eingesetzt wird oder die Mahalanobis-Distanz (MM) eingesetzt. Für beide Metriken zeigte sich, dass der proportional noch sehr hohe Anteil von Merkmalen, die nicht für die Klassifikation von Bedeutung sind, Schwierigkeiten verursacht. So zeigt die Tanimoto-Distanz nur für bereits geeignet reduzierte Merkmalsets gute Ergebnisse und auch für die Mahalanobis-Distanz lässt sich solch ein Effekt feststellen [VSH06b]. Klinische Datensätze komplexer Krankheitszustände sind häufig durch Multimodalität ge-

kennzeichnet. Diesem Aspekt Rechnung tragend wurden die betrachteten prototype basierten Verfahren um lokale Metrikadaptation erweitert, analysiert und zur Klassifikation von Proteomdaten eingesetzt. Die damit ermittelten lokalen Klassifikationsmodelle (angezeigt durch L-) sind besonders für klinische Fragestellungen geeignet, bei denen globale Modellierungen inadequat sind, wie z.B. bei Patientengruppen mit vielfältigen Krankheitsbildern.Die Ergebnisse für einige der betrachteten Datensätze sind in Tabelle (1) zusammengefasst. Die Ergebnisse zeigen Verbesserungen in der Vorhersagegenauigkeit im

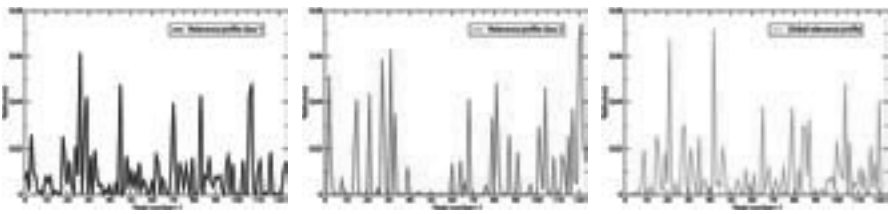

Abbildung 3: Relevanzprofil für Klasse 1, 2 und global für Proteom II mit dem LSNPC Verfahren

Vergleich zu NPC-Verfahren, bei denen die Metrik nicht adaptiert wurde und auch im Vergleich zu SRNG. Bei den betrachteten Datensätzen konnte keine signifikanten Verbesserung bei lokaler im Vergleich zu globaler Metrikadaptation erkannt werden, allerdings sind die ermittelten nun lokalen Relevanzbewertungen interessant, um zusätzliche Informationen über klassenspezifische Attribute zu erhalten. Die lokalen Relevanzprofile für den Proteom II Datensatz sind in Abbildung 3 dargestellt. Dabei wird deutlich das einige Peaks (105,43) sowohl im globalen als auch in den lokalen Profilen, allerdings dort in exklusiver Form auftreten. Dieser Ansatz kann in Bezug zu klassischen post-hoc Tests gebracht werden. Allerdings sollten die Relevanzprofile nicht überinterpretiert werden, da diese zwischen verschiedenen Läufen des Verfahrens variieren können und die Einzeldimensionen als unabhängig betrachtet werden, aber Peaks aufgrund der darunterliegenden Peptidfragmentierung korreliert sein können.

6.2 Überwachtes Lernen mit Fuzzy-Label-Information

Nachfolgend werden Varianten des FSNPC-Algorithmus betrachtet und mit verschiedenen Metriken für die Datensätze analysiert. Die Ergebnisse sind in Tabelle 1 dargestellt und zeigen vergleichbare Ergebnisse, dabei zeigt sich auch, dass bei Verwendung lokaler Metrikadaption keine weitere Verbesserung eintritt. Bei der Analyse der Fuzzy-Methoden wird eine Verbesserung von Erkennungs und Vorhersageleistung der Verfahren sichtbar, wenn die verwendete Metrik adaptiert wird. Unter Berücksichtigung der Fuzzy-Labels der Prototypen erkennt man, dass das Verfahren in der Lage war die Klassenbezeichner der Trainingsdaten korrekt zu lernen. FSNPC generiert Prototypen mit sehr deutlichen Klassenlabeln aber auch sehr unscharfe Klassenzuordnung, die Split-Entscheidungen anzeigen. Insbesondere letzteres ist bedeutsam, um zu erkennen, dass Zuordnungen von neuen Datenpunkten zu derartigen Prototypen eine unsichere Klassifikationen bedeuten. Im zwei Klassenfall können die per Relevanzlernen ermittelten relevanten Peaks zudem durch eine ROC-Analyse genauer bzgl. ihrer Sensitivität und Spezifizität analysiert werden. Die vier Peaks $(43, 105, 121, 123)$ zeigen deutlich ein unterschiedliches Verhalten bzgl. ihrer Verwendbarkeit für eine klinische Analyse. Bei Betrachtung der AUC (Fläche unter der Kurve) Werte wird deutlich, dass insbesondere der Peak 43 besonders nützlich ist da er so-

Methode	SNG	SNG	SNG	GLVQ	GLVQ	SNPC	SNPC	SNPC	FSNPC	FSNPC	FSNPC	QDA	LDA	SVM
Metrik	Euc	Euc_λ	Euc_{λ_r}	TM	MM	Euc	Euc_λ	Euc_{λ_r}	Euc	Euc_λ	Euc_{λ_r}	MM	MM	Poly
WDBC	95	97	96	96	89	89	95	97	93	91	95	94	96	87
Proteom_1	78	84	76	78	78	79	87	87	70	87	87	75	81	76
Proteom_2	87	93	85	65	90	82	87	91	92	93	91	71	81	85

Tabelle 1: Modellgenauigkeiten in $\%$ für die verschiedenen Datensätze, mit den Metriken Euklidisch(Euc), skaliert Euklidisch (Euc_λ), lokal skaliert Euklidisch (Euc_{λ_r}), Mahalanobismetrik(MM), Tanimotometrik (TM) und einem Polynomialkernel (Poly) mit $\gamma = \frac{1}{1000}, r = 1.0, d = 2.0$.

wohl hohe Sensitvität als auch Spezifizität aufweist. Somit können mit der ROC-Analyse die identifizierten Peaks in einer strengeren klinisch, univariaten Sicht beurteilt und selektiert werden. Dies ist ein üblicher Nachfolgeschritt um eine klinisch verwendbare Lösung von den identifizierten Markerkandidaten zu finden. Die verschiedenen Ergebnisse bei Verwendung verschiedener Klassifikatoren und Datensätze sind in Tabelle 1 dargestellt. Nahezu alle Klassifikatoren zeigen gute Ergebnisse von über $> 90\%$ für den WDBC Datensatz [BM06]. Der Proteom_1 Datensatz ist der komplizierteste Datensatz und wird am besten durch einen Klassifikator mit Relevanzlernen modelliert. Der Proteom_2 Datensatz wird durch mehrere Klassifikatoren korrekt modelliert allerdings, führt auch hier Relevanzlernen zu einer weiteren Verbesserung. Klassifikationsverfahren mit Fuzzy-Labeling zeigen ähnlich gute Ergebnisse führen aber zu einer einfacheren Modellgenerierung im Bezug auf die Netzwerkstruktur. Die lokale Metrikadaptation hat für die betrachteten Datensätze nur leichte Vorteile. Die abschliessende Analyse zeigt, dass die durchgeführten Erweiterungen von prototypbasierten Methoden erfolgreich für reale Datensätze aus der klinischen Proteomik einsätzbar sind. Dabei ist die Verwendung von Metrikadaptation nützlich, um potentielle Biomarkerkandidaten zu identifizieren sowie, um die Vorhersagegenauigkeit zu verbessern. Lokales Relevanzlernen verbesserte die Ergebnisse im allgemeinen nicht, aber gestattete eine spezifischere Interpretation der relevanten Peaks. Der fuzzy SNPC Algorithmus gestattet Sicherheitsbewertungen der Klassifikationsentscheidungen. Zusätzlich konnte eine Verbesserung der Vorhersagegenauigkeit festgestellt werden. Dies ist vorallem auf eine flexiblere Modellierung des Klassifikators zurückzuführen, da die Klassenbezeichner der Prototypen nicht länger fest vorgegeben sind und somit nur die allgemeine Anzahl der Prototypen spezifiziert werden muss. Die Tanimoto und Mahalanobisdistanz sind für Daten der klinischen Proteomik nur nach geeigneter Vorselektion der Merkmale anwendbar. Ähnliche Ergebnisse konnten für LDA und QDA festgestellt werden. Im Vergleich zum bekannten SVM-Algorithmus sind die Ergebnis bei Verwendung von Prototypenverfahren ähnlich gut oder zum Teil auch besser, aber mit dem zusätzlichen Vorteil verbesserter Interpretierbarkeit.

Literatur

[BM06] C. Blake und C. Merz. UCI repository of machine learning databases., (last visit 01.11.2006). available at: http://www.ics.uci.edu/ mlearn/MLRepository.html.

[CGBAA02] K. Crammer, R. Gilad-Bachrach, A.Navot und A.Tishby. Margin analysis of the LVQ algorithm. In *Proc. NIPS 2002*, http://www-2.cs.cmu.edu/Groups/NIPS/NIPS2002/NIPS2002preproceedings/index.html, 2002.

[HSV05a] B. Hammer, M. Strickert und Th. Villmann. On the generalization ability of GRLVQ networks. *Neural Processing Letters*, 21(2):109–120, April 2005.

[HSV05b] B. Hammer, M. Strickert und Th. Villmann. Supervised Neural Gas with General Similarity Measure. *Neural Processing Letters*, 21(1):21–44, 2005.

[HTF01] T. Hastie, R. Tibshirani und J. Friedman. *The Elements of Statistical Learning*. Springer, New York, 2001.

[HV02] B. Hammer und Th. Villmann. Generalized Relevance Learning Vector Quantization. *Neural Networks*, 15(8-9):1059–1068, 2002.

[HV03] B. Hammer und Th. Villmann. Mathematical Aspects of Neural Networks. In M. Verleysen, Hrsg., *Proc. Of European Symposium on Artificial Neural Networks (ESANN'2003)*, Seiten 59–72, Brussels, Belgium, 2003. d-side.

[Koh95] Teuvo Kohonen. *Self-Organizing Maps*, Jgg. 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 1995. (2nd Ext. Ed. 1997).

[Lie02] Daniel C. Liebler. *Introduction to Proteomics*. Humana Press, 2002.

[PKH06] J. Prados, A. Kalousis und M. Hilario. On Preprocessing of SELDI-MS data and its evaluation. In *International Symposium on Computer Based Medical Systems (CBMS 2006)*, Seiten 953–958. IEEE press, 2006.

[SBO03] S. Seo, M. Bode und K. Obermayer. Soft nearest prototype classification. *IEEE Transaction on Neural Networks*, 14:390–398, 2003.

[Sch06] Frank-Michael Schleif. *Prototype based Machine Learning for Clinical Proteomics*. Technical University Clausthal, 2006. Dissertation.

[SS04] A.C. Sauve und T.P. Speed. Normalization, Baseline Correction and Alignment of High-Throughput Mass Spectrometry Data. In *Proceedings Gensips*, 2004. to be published, preprint http://stat-www.berkeley.edu/users/terry/Group/publications/Final2Gensips2004Sauve.pdf.

[SY95] A. S. Sato und K. Yamada. Generalized Learning Vector Quantization. In G. Tesauro, D. Touretzky und T. Leen, Hrsg., *Advances in Neural Information Processing Systems*, Jgg. 7, Seiten 423–429. MIT Press, 1995.

[VSH06a] T. Villmann, F.-M. Schleif und B. Hammer. Prototype-based fuzzy classification with local relevance for proteomics. *Neurocomputing Letters*, Seiten 2425–2428, 2006.

[VSH06b] Th. Villmann, F.-M. Schleif und B. Hammer. Comparison of Relevance Learning Vector Quantization with other Metric Adaptive Classification Methods. *Neural Networks*, 19:610–622, 2006.

[WNP+04] M. Wagner, D. Naik, A. Pothen, S. Kasukurti adn R. Devineni, B. Adam, O. Semmes und G. Wright. Computational protein biomarker prediction: a case study for prostate cancer. *BMC Bioinformatics*, 5(26):open access, 2004.

Frank-Michael Schleif wurde am 11. Dezember 1977 in Leipzig geboren. Bis 2002 studierte er Angewandte Informatik an der Universität Leipzig. Danach arbeitete er als wissenschaftlicher Mitarbeiter am Lehrstuhl für Angewandte Telematik der Universität Leipzig bevor er eine Industriepromotion bei der Firma Bruker Bioscience Corp., mit Betreuung an der Universität Clausthal aufnahm. Während der Arbeit bei Bruker war er in verschiedenen Forschunsprojekten und der Software-Entwicklung für den Bereich klinische Proteomik tätig. Die Ergebnisse der Arbeit wurden in einer Vielzahl von Konferenz- und Journalpublikationen vorgestellt und fanden Eingang in das kommerzielle Datenanalysepaket ClinProTools. Im Dezember 2006 wurde die Dissertation an der Technischen Universität Clausthal mit *magna cum laude* verteidigt. Seitdem arbeitet er an der Universität Leipzig in der AG Computational Intelligence in Projekten der Datenanalyse. Seine Forschungsinteressen liegen im Bereich Mustererkennung, statistische Datenanalyse und Algorithmen Entwicklung.