

4.1.2 Klassifikation von Proteinen

Karchin, Karplus, Haussler (2002): Classifying G-protein coupled receptors with support vector machines. *Bioinformatics* 18.1, pp. 147–159

GPCRs bilden superfamilie von Membran-Proteinen.

Aufgabe: unterscheide mehrere Familien bzw. subfamilien, z.B. finde solche die am selben Liganden binden.

Unterscheidung von k Gruppen durch k one-to-rest-classifier:

$$h(x) = \arg \max_k \sum_{i=1}^l y_i^k \xi_i^k K_k(x_i, x)$$

Eingabeparameter für die Proteine: Konstruiere für GPCRs profilHMM und berechne für gegebenes Protein X für jeden Zustand s und jede Aminosäure x

$$f_s(x) = \frac{\partial}{\partial \theta_{x,s}} \log \Pr(X \mid \theta, \tau) = \frac{\zeta(x, s)}{\theta_{x,s}} - \zeta(s)$$

Dabei sind $\theta_{x,s}$ die Emissionsw'keiten, τ die Übergangsw'keiten, $\zeta(x, s)$ und $\zeta(s)$ die erwarteten Anzahlen der Emissionen von x aus dem Zustand s bzw. der Besuche in s .

Die $f_s(x)$ (oder gewichtete Summen von diesen) werden für X in die SVM eingegeben.

SVMtree: zur Beschleunigung wurden GPCRs zunächst in 5 Klassen eingeteilt und diese dann weiter unterteilt. Proteine werden dann hierarchisch zugeordnet.

Verglichene Methoden: SVMs mit Gauß-Kern, BLAST, profileHMM, kernNN

Ergebnisse: HMMs haben GPCRs am besten erkannt, SVM konnten die subfamilien am besten erkennen.

Ähnlicher Ansatz, zu einem probabilistischen Modell, z.B. HMM, gegeben durch $\text{Pr}_\theta(\cdot)$ mit $\theta \in \mathbb{R}^p$, einen Kernel zu bauen:

Sei $\ell_x(\theta) := \log \text{Pr}_\theta(x)$, $d\ell_x(\theta) = \left(\frac{\partial \ell_x(\theta)}{\partial \theta_1}, \dots, \frac{\partial \ell_x(\theta)}{\partial \theta_p} \right)$ und $J(\theta)$ die Fisher-Informationsmatrix, d.h.

$$J_{ij}(\theta) := \mathbb{E} \left(\frac{\partial \ell_X(\theta)}{\partial \theta_i} \cdot \frac{\partial \ell_X(\theta)}{\partial \theta_j} \right) = \mathbb{E} \left(-\frac{\partial^2}{\partial \theta_i \partial \theta_j} \ell_X(\theta) \right)$$

Sei $\hat{\theta}$ ein aus einem Lerndatensatz geschätzter Wert für θ . Dann definieren Jaakola und Haussler (1999) den *Fisher-Kernel* durch

$$K(x, y) := d\ell_x(\hat{\theta}) \cdot J(\hat{\theta})^{-1} \cdot d\ell_y(\hat{\theta})^T.$$

Hua, Sun (2001): Support vector machine approach for protein subcellular localization prediction. *Bioinformatics* 17.8, pp. 721–728

Aufgabe: Schätze den Ort eines Proteins in der Zelle anhand der anteilmäßigen Aminosäure-Zusammensetzung.

Klassen bei Prokaryonten: Cytoplasma, Periplasma, Extrazellulär.

Klassen bei Eukaryonten: Cytoplasma, Extrazellulär, Mitochondrial, Zellkern

Eingabe: Aminosäurehäufigkeiten im Protein. Eingabe für ein Protein ist also ein $x \in \mathbb{R}^{20}$.

multiple Klassifikation mit one-versus-rest

Jackknife zur Schätzung der Vorhersagegenauigkeit, bewertet nach korrekt geschätzter Klassenzugehörigkeiten und Matthew's Correlation Coefficient.

Wieder schneiden die SVMs recht gut ab, die Wahl der Kernels und der Parameter haben geringen Einfluß aufs Ergebnis.

Garg, Bhasin, Raghava (2005) SVM-based Method for Subcellular Localization of Human Proteins Using Amino Acid Compositions, Their Order, and Similarity Search *Journal of Biological Chemistry*, 280(15), 14427–14432

Aus SWISSPROT 3780 Proteine ausgewählt, die untereinander nicht zu ähnlich sind, und zu 4 Lokalisationen gehören: Cytoplasma, Mitochondrium, Nucleus, Plasma-Membran.

1-versus-rest-SVM, freie Software SVM_light verwendet;

Verwendete Eigenschaften von Proteinen:

Kombinationen von:

Aminosäure-Zusammensetzung: Vektor $\in \mathbb{R}^{20}$

Häufigkeiten von Dipetiten: Vektor $\in \mathbb{R}^{400}$

Häufigkeiten von Aminosäure-Paaren mit Abstand ≤ 4 : Vektor $\in \mathbb{R}^{1600}$

PSI-BLAST-Lokalisations-Vorhersage, dargestellt in $\in \mathbb{R}^5$ mit einer 1 und vier Nullen ((0,0,0,0,1) ist “unknown”).

Beste Kombination: 450-dimensionaler Vektor, bestehend aus Aminosäure- und Dipeptid-Häufigkeiten und PSI-BLAST-Ausgabe zusammen mit Gauß-Kern und Soft-Margin liefert fast 85% Genauigkeit. (PSI-BLAST allein 73,3%)

(Ermittelt mit 5-facher Kreuz-Validierung)

HSLPred: Verschiedene Varianten von SVMs werden auf die Daten angewandt. Aus den Ergebnissen schätzt eine weitere SVM die Lokalisation.

Hua, Sun (2001) A novel method of protein secondary structure prediction with high segment overlap measure: Support Vector Machine approach.

***J. Mol. Biol.* 308, pp.397-407** vorherzusagen: Helix (H), Faltblatt (F), Coil (C)

Dateneingabe: Element aus $\{0, 1\}^{21 \cdot l}$ für Fenster der Länge l . Aminosäure und Start/Stop-Zeichen werden jeweils durch eine 1 und zwanzig Nullen dargestellt.

Gauß-Kern für binäre Klassifikationen H/ \neg H, F/ \neg F, C/ \neg C, H/F, C/F, C/H

Fenstergrößen zwischen 5 und 17 Positionen erprobt, optimal 13, aber keine großen Unterschiede

Klassifikation H/F/C aus binären Klassifikationen:

SVM_MAX_D: größte pos. Distanz unter $H/\neg H$, $F/\neg F$, $C/\neg C$ zur trennenden Ebene

SVM_TREE1: Erst $H/\neg H$ und im Fall von $\neg H$ folgt F/C .

SVM_TREE2, **SVM_TREE3**: analog mit veränderter Reihenfolge

SVM_VOTE: Abstimmung unter allen binären Klassifikationen

SVM_NN: Eingabe der 6 binären in Neuronales Netz mit verdeckter Schicht aus 20 Neuronen.

SVM_JURY: Kombination aus den oben genannten.

Ergebnisse

Bei bestimmten Datensätzen geringfügig besser als bisher übliche Methoden

SVM_JURY am besten, SVM_MAX_D am zweitbesten

Ward, McGuffin, Buxton, Jones (2003) Secondary structure prediction with support vector machines *Bioinformatics* 19.3, pp.1650–1655

Aufgabe: Unterscheide Helix (H), Faltblatt (F) und Coil (C)

Eingabe: Positionsspezifische PSI-BLAST-Score-Matrizen für Sequenzfenster der Länge 15 um jeweilige Position.

Daten für Training und Validierung: 1460 nicht-redundante Proteine ($\leq 25\%$ Übereinstimmung zwischen je zwei).

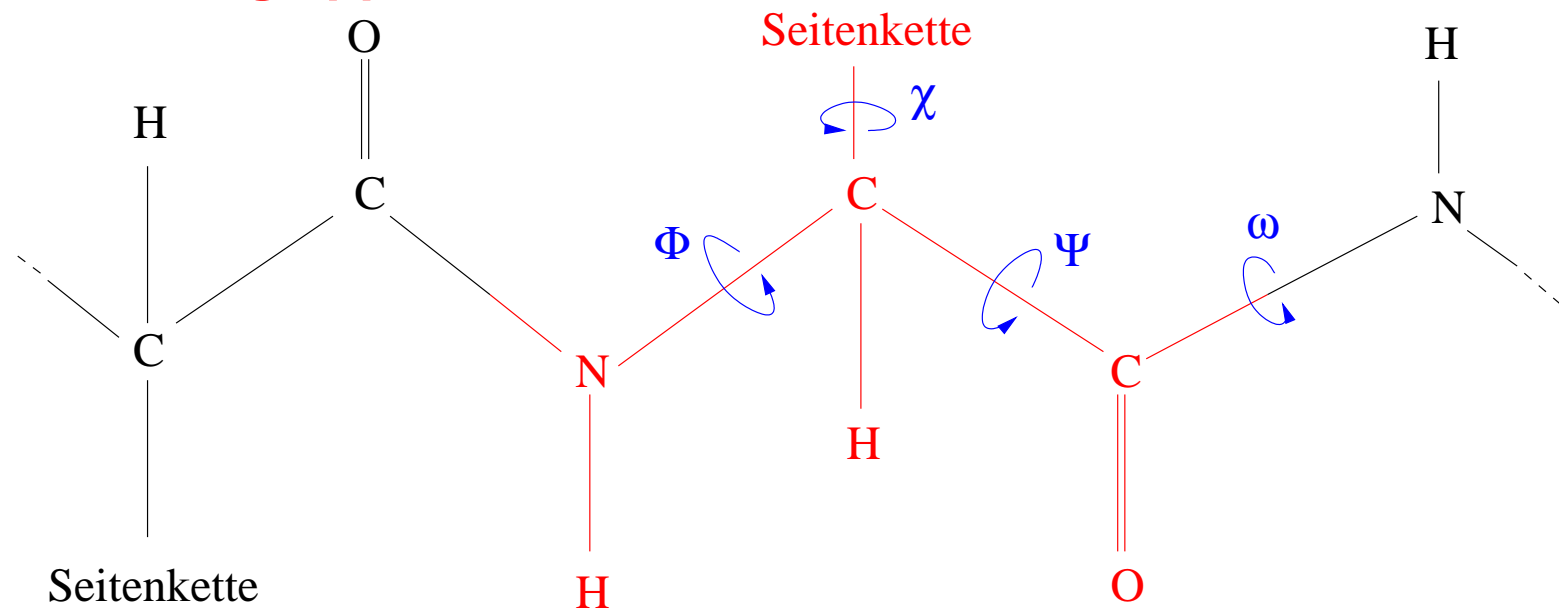
Kern für H/ \neg H, F/ \neg F, C/ \neg C, H/F, C/F, C/H:

$$K(x, y) = \left(\frac{\langle x, y \rangle + 1}{50} \right)^2$$

Testsieger; heuristische Erklärung: berücksichtigt periodische Abhängigkeiten

Beste Ergebnisse bei Kombination von SVM mit PSIPRED ('state of the art',
Neuronales Netz) und PROFsec

Kuang, Leslie, Yang (2004) **Protein backbone angle prediction with machine learning approaches.** *Bioinformatics*



Problem: Schätze alle Φ und Ψ aus Sequenz

Es werden 5 Klassen von Winkelpaaren (Φ , Ψ) definiert, gemäß Vorkommen in Sekundärstrukturen. Sequenzen sollen diesen 5 Klassen zugeordnet werden.

Vergleich SVM vs. Neuronales Netz (NN)

Eingabe: Zur Schätzung von (Φ , Ψ) einer Position wird Teilsequenz der Länge 9 eingegeben.

Bei NN: PSI-BLAST-Profiles, LSBSP1-Datenbank enthält 138604 Positionsspezifische Score-Matrizen.

Bei SVM: drei verschiedene Ansätze

drei Ansätze für Eingabe in SVM:

- Binary encoding feature map: Für jede Position Vektor mit zwanzig Nullen und einer 1. (21. Eintrag steht für nichtvorhandene Aminosäure vor Anfang und nach Ende)
- PSI-BLAST Profile Feature Map
- Predicted Secondary structure feature map: PSI-PRED
Sekundärstrukturvorhersagen für Fragmente der Länge 9.

Kern: Linear! (Gauß-Kern brachte nur sehr geringe Verbesserung)

Ergebnisse: gute Vorhersagen in Helix- und Faltblatt-, schlechte in Coil-Regionen

SVM etwas besser als NN

Structure Feature etwas besser als profile feature, besser als binary.

Noch besser: Kombinationen, insbes. mit **wahrer** Sekundärstruktur.

Zhang, Yoon, Welsh (2005) Improved method for predicting β -turn using SVM *Bioinformatics* 21(10): 2370–2374

Vergleich von SVM-Methode zur β -Schleifen-Vorhersage mit älteren Verfahren, zB. NeuralNetwork-basiertes BetaTPred2.

7-fach Kreuzvalidierung mit 426 nicht-homologen Proteinen.

SVM mit Gauß-Kern Testsieger mit ca. 75% Genauigkeit.

Verwenden freie Software SVMlight

Verschiedene Ansätze für einzelne Sequenzen und Sequenz-Profile (nach Benutzung von PSI-BLAST)

Single Sequence: Für je 7 benachbarte Positionen erhalten wir ein $\in \{0, 1\}^{140}$, mit 7 Einsen, die für die entsprechende AA an der entsprechenden Position stehen.

Profiles: Für jede der 7 Positionen wird der von PSI-BLAST generierte Score-Vektor mit jeweils

$$x \mapsto \frac{1}{1 + e^{-x}}$$

auf $[0, 1]^{20}$ abgebildet. Wir erhalten also ein $\in [0, 1]^{140}$.

Außerdem: PSIPRED-Vorhersage codiert durch (1, 0, 0) (Helix), (0, 1, 0) Faltblatt, (0, 0, 1) Coil.

Vert (2002) A tree kernel to analyze phylogenetic profiles. *Bioinformatics*

1.1, pp. 1–9 Ein phylogenetisches Profil eines Proteins ist die Information in welchem der bisher Sequenzierten Genome das Protein vorkommt und in welchen nicht.

Wir denken uns die Genome als Blätter eines Stammbaums und die Profile ordnen dann den Blättern jeweils 0 oder 1 zu.

Problem: Finde Proteine mit ähnlichen Profilen, denn die haben vielleicht was miteinander zu tun. Wie messen wir die Ähnlichkeit von Proteinen?

$$K_{\text{tree}}(x, y) := \sum_S \sum_{z_S} p(z_S) \cdot p(x|z_S) \cdot p(y|z_S)$$

dabei ist S ein Teilbaum der Phylogenie, der die Wurzel enthält und z_S ist eine Beschriftung der Knoten in S mit 0 und 1. p steht für Wahrscheinlichkeit.

Summiert wird also jeweils die W'keit, dass die Proteine der Profile x und y gemäß z_S gemeinsam und dann getrennt evolviert sind.

Modellparameter: Für jede Kante Übergangswahrscheinlichkeiten $0 \leftrightarrow 1$, W'keit für 1 in der Wurzel.

Berechnung von $K_{\text{tree}}(x, y)$ mittels dynamischer Programmierung.

Verglichen wird mit dem "naiven" Kern $K_{\text{naiv}}(x, y) = \sum_u x_u \cdot y_u$.

Bewertung der Kernel: Kreuzvalidierung mit (true positives)/(false positives) für 16 verschiedene bekannte Funktionsklassen.

K_{tree} schneidet etwas besser ab als K_{naiv} .

Außerdem im Paper: KernelPCA für 4 funktionelle Familien.

(Hauptkomponentenanalyse im Feature-Space. Wie bei Standard-SVMs muss nicht wirklich im Feature-Space sondern nur mit dem Kernel gerechnet werden.)

5 Wieviele Beispiele brauchen wir um unendlich viele Hypothesen zu unterscheiden?

5.1 Die Vapnik-Chervonenkis-Dimension (VC)

Wir kommen zurück auf's PAC-Modell. Zur Erinnerung:

PAC = Probably Approximately Correct

Ziel: Lerne aus Beispielen, so dass mit einer Wahrscheinlichkeit von mindestens $1 - \delta$ der Fehler beim Klassifizieren höchstens ϵ ist.

Ist D eine W'keitsverteilung auf Ω und $C \in \mathcal{C}$ das zu lernende Konzept, so können wir den **Fehler** der Hypothese $H \in \mathcal{H}$ durch

$F_D(C, H) := D((C \cup H) \setminus (C \cap H))$ definieren.

Fairness-Bedingung: Bewerte den Fehler mit derselben Verteilung D , gemäß der die Lernbeispiele erzeugt wurden.

PAC-Algorithmus für eine Konzeptklasse \mathcal{C}

Ablauf:

- Eingabe: Vertrauensparameter δ , Fehlerparameter ϵ , Länge n der Beispiele
- Bestimme Anzahl $s = s(\delta, \epsilon, n)$ der anzufordernden Beispiele.
- Lies s Beispiele an
- Gib ein $H \in \mathcal{H}$ aus.

Kriterium: Für alle $\epsilon, \delta \in (0, 1]$, $C \in \mathcal{C}$ und alle Verteilungen D auf Ω gilt:

$$W_{sD}(F_D(C, H) \leq \epsilon) \geq 1 - \delta$$

($W_{sD}(\cdot)$:= W'keit unter der Voraussetzung, dass die Lerndaten aus D kommen)

Ein PAC-Algorithmus für endliches $|\mathcal{H}|$:

Falls $\mathcal{H} = \mathcal{C}$ und es leicht ist, für jede mit $C \in \mathcal{C}$ verträgliche Folge von Beispielen $\{(x_1, b_1), \dots, (x_s, b_s)\} \subset \Omega \times \{0, 1\}$ (d.h. $b_i = 1 \Leftrightarrow x_i \in C$) eine konsistente Hypothese $H \in \mathcal{H}$ zu finden (d.h. $b_i = 1 \Leftrightarrow x_i \in H$), dann

- setze $s := \lceil (\ln(|\mathcal{C}|) - \ln(\delta)) / \varepsilon \rceil$
- lies s Beispiele ein
- Gib dazu konsistente Hypothese H aus

Brauchen wir also unendlich viele Beispiele falls $|\mathcal{H}| = \infty$?

Definition: Die Konzeptklasse \mathcal{C} **zertrümmert** die Menge S , falls

$$\mathcal{P}(S) = \{S \cap C \mid C \in \mathcal{C}\}.$$

Die **Vapnik-Chervonenkis-Dimension** $VC(\mathcal{C})$ von \mathcal{C} ist die Mächtigkeit der größten Menge S , die von \mathcal{C} zertrümmert wird.

Beispiele:

- Die Klasse der Halbräume im \mathbb{R}^n hat VC-Dim $n + 1$
- Die Klasse der achsenparallelen Rechtecke in $\mathbb{Z}^d \cap [-2^n, 2^n]$ hat VC-Dim $2d$.
- Die Klasse der $U \subset \mathbb{R}$ mit $\exists \alpha : U = \{x : \sin(x\alpha) > 0\}$ hat VC-Dim ∞ , denn sie zerschmettert die Menge $\{10^{-i} \mid i = 1, \dots, n\}$ für jedes n .

Satz 15 Sei $2 \leq VC(\mathcal{C}) < \infty$, $\varepsilon < 1/4$ und $\delta < 1/2$. Dann muss der entsprechende PAC-Algorithmus mindestens

$$\Omega\left(\frac{1}{\varepsilon} \left[VC(\mathcal{C}) + \ln\left(\frac{1}{\delta}\right) \right]\right)$$

Beispiele anfordern.

Beweis: Zeige zunächst $\Omega(\text{VC}(\mathcal{C})/\varepsilon)$.

Konstruiere schwierige Verteilung \mathcal{D} :

\mathcal{C} mit $\text{VC}(\mathcal{C}) = d + 1$ zertrümmere $S = (x_0, \dots, x_d)$.

$$\mathcal{D}(x) = \begin{cases} 0 & \text{für } x \notin S \\ 1 - 4\varepsilon & \text{für } x = x_0 \\ 4\varepsilon/d & \text{für } x \in S \setminus x_0 \end{cases}$$

\Rightarrow Konzeptklasse $\{S \cap C \mid C \in \mathcal{C}\}$.

Bei weniger als $d/(16\varepsilon)$ Bsp. sehen wir (wahrscheinlich) mehr als $d - 4\varepsilon d/(16\varepsilon) = 3d/4$ Bsp. nicht und klassifizieren davon (wahrscheinlich) mehr als $3d/8$ falsch.

\Rightarrow Fehler $\geq 3d/8 \cdot 4\varepsilon/d = 3/2\varepsilon$.

Zeige noch, dass mehr als $\frac{(1-\varepsilon)}{\varepsilon} \ln(1/\delta)$ Beispiele nötig...

□

Satz 16 *Es gibt einen PAC-Algorithmus, der höchstens*

$$\left\lceil \frac{4}{\varepsilon} \cdot \left(VC(\mathcal{C}) \cdot \ln \left(\frac{12}{\varepsilon} \right) + \ln \left(\frac{2}{\delta} \right) \right) \right\rceil$$

anfordert.

Beweisschritte: S sei Beispielmenge gemäß Verteilung D .

Menge der ε -Fehlerregionen von $C \in \mathcal{C}$ sei

$$\Delta_\varepsilon(C) := \{C \cup C' \setminus C \cap C' \mid C' \in \mathcal{C}, F_{\mathcal{D}}(C, C') > \varepsilon\}$$

$$A := \{\exists U \in \Delta_\varepsilon(C) : S \cap U = \emptyset\}$$

zu zeigen: $\Pr(A) \leq \delta$

mit Hilfe von Gedankenexperiment: Sei S' eine weitere Beispielfolge mit $|S'| = |S| =: s$.

$$B := \{\exists U \in \Delta_\varepsilon(C) : S \cap U = \emptyset \wedge |U \cap S'| \geq \varepsilon \cdot s/2\}$$

Zeige mit Hilfe von Chernoff-Schranke:

Für $s \geq 8/\varepsilon$ gilt $\Pr(B) = \Pr(A) \cdot \Pr(B|A) \geq \Pr(A) \cdot \left(1 - e^{-\left(\frac{1}{2}\right)^2 \frac{\varepsilon \cdot s}{2}}\right) \geq \frac{1}{2} \Pr(A)$

Dann bleibt zu zeigen: $\Pr(B) \leq \delta/2$.

$$\begin{aligned} \Pr(B) &\leq \sum_U \Pr\left(U \cap S = \emptyset \wedge |U \cap S'| \geq \frac{\varepsilon \cdot s}{2}\right) \\ &\leq \mathbb{E}|\{U = (S \cup S') \cap C \mid C \in \mathcal{C}\}| \cdot \left(\frac{1}{2}\right)^{\frac{\varepsilon s}{2}} \end{aligned}$$

Um daraus $\Pr(B) \leq \delta/2$ zu zeigen, verwenden wir folgendes

Lemma 1

$$|\{C \cap S \mid C \in \mathcal{C}\}| \leq \sum_{i=0}^{\text{VC}(\mathcal{C})} \binom{|S|}{i} \begin{cases} = 2^{|S|} & \text{für } |S| < \text{VC}(\mathcal{C}) \\ \leq \left(\frac{e \cdot |S|}{\text{VC}(\mathcal{C})}\right)^{\text{VC}(\mathcal{C})} & \text{für } |S| \geq \text{VC}(\mathcal{C}) \geq 1 \end{cases}$$

Damit folgt nämlich $\Pr(A) \leq 2 \Pr(B) \leq 2 \cdot \left(\frac{e \cdot 2s}{\text{VC}(\mathcal{C})}\right)^{\text{VC}(\mathcal{C})} \cdot e^{-\varepsilon s/2}$.

Bleibt also zu zeigen, dass gilt $\left(\frac{e \cdot 2s}{\text{VC}(\mathcal{C})}\right)^{\text{VC}(\mathcal{C})} \cdot e^{-\varepsilon s/2} \leq \delta/2$

oder äquivalent:

$$\text{VC}(\mathcal{C}) \cdot \ln \left(\frac{e \cdot 2s}{\text{VC}(\mathcal{C})} \right) - \frac{\varepsilon s}{2} \leq \ln \frac{\delta}{2}$$

Nach Anwendung diverser Umformungen und Ungleichungen für Logarithmen zeigt sich, dass dafür folgendes genügt:

$$s \geq \frac{4}{\varepsilon} \cdot \left(\text{VC}(\mathcal{C}) \cdot \ln \left(\frac{12}{\varepsilon} \right) + \ln \left(\frac{2}{\delta} \right) \right)$$

□

Beweis des Lemmas: Sei $\Pi_{\mathcal{C}}(S) := \{C \cap S \mid C \in \mathcal{C}\}$.

erste Ungleichung: Induktion über $|S|$: Sei $|S| = m$ und die Induktionsannahme ist, dass die Aussage

$$|\Pi_{\mathcal{C}}(S')| \leq \sum_{i=0}^{\text{VC}(\mathcal{C})} \binom{|S'|}{i}$$

für alle S' mit $|S'| < m$ gilt, insbesondere also für $S' := S \setminus \{x\}$ (für irgendein gewähltes $x \in S$).

Sei

$$\mathcal{C}' := \{U \in \Pi_{\mathcal{C}}(S) \mid x \notin U, U \cup \{x\} \in \Pi_{\mathcal{C}}(S)\}$$

Dann gilt $\text{VC}(\mathcal{C}') < \text{VC}(\mathcal{C})$ (wegen $\text{VC}(\Pi_{\mathcal{C}}(S)) < \text{VC}(\mathcal{C})$) und

$$\begin{aligned} |\Pi_{\mathcal{C}}(S)| &= |\Pi_{\mathcal{C}}(S \setminus \{x\})| + |\Pi_{\mathcal{C}'}(S \setminus \{x\})| \\ &\leq \sum_{i=0}^{\text{VC}(\mathcal{C})} \binom{|S| - 1}{i} + \sum_{i=0}^{\text{VC}(\mathcal{C}) - 1} \binom{|S| - 1}{i} \\ &= \sum_{i=0}^{\text{VC}(\mathcal{C})} \binom{|S|}{i} \end{aligned}$$

zweite Ungleichung: Für $m < d$ gilt $\sum_{i=0}^d \binom{m}{i} = \sum_{i=0}^m \binom{m}{i} = 2^m$, für $m \geq d \geq 1$ hingegen:

$$\begin{aligned} \left(\frac{d}{m}\right)^d \cdot \sum_{i=0}^d \binom{m}{i} &\leq \sum_{i=0}^m \left(\frac{d}{m}\right)^i \cdot \binom{m}{i} \\ &= \left(1 + \frac{d}{m}\right)^m \\ &\leq e^d \end{aligned}$$

und damit

$$\sum_{i=0}^d \binom{m}{i} \leq \left(\frac{e \cdot m}{d}\right)^d$$

□

5.2 Hard Margins

Als Analogon zur VC-Dimension definieren wir:

Definition Sei \mathcal{F} eine Familie von Funktionen $f : X \rightarrow \mathbb{R}$ und $\gamma > 0$. Die Menge $S = \{x_1, \dots, x_m\} \subset X$ wird **γ -zertrümmert (*shattered*) durch \mathcal{F}** , falls es einen Vektor r gibt, so dass für jeden “Klassifizierungsvektor” $b \in \{-1, 1\}^m$ eine Funktion $f_b \in \mathcal{F}$ existiert, so dass gilt

$$f_b(x_i) \begin{cases} \geq r_i + \gamma & \text{falls } b_i = 1 \\ \geq r_i - \gamma & \text{falls } b_i = -1 \end{cases}$$

Die **Fat-Shattering-Dimension $\text{fat}_{\mathcal{F}}(\gamma)$ von \mathcal{F} für γ** ist dann das Maximum von $|S|$ über alle S , die durch \mathcal{F} γ -zertrümmert werden.

Definition Sei \mathcal{F} eine Familie von Funktionen $f : X \rightarrow \mathbb{R}$ und $\gamma > 0$. Eine **γ -Überdeckung** von \mathcal{F} für $S = \{x_1, \dots, x_m\} \subset X$ ist eine endliche Familie $B(S)$ von Funktionen mit:

$$\forall f \in \mathcal{F} \exists g_f \in B(S) \forall x \in S : |f(x) - g_f(x)| < \gamma$$

Die **Überdeckungszahl** $\mathcal{N}(\mathcal{F}, S, \gamma)$ von \mathcal{F} für S ist die Größe der kleinstmöglichen γ -Überdeckung und die **Überdeckungszahl** $\mathcal{N}(\mathcal{F}, m, \gamma)$ von \mathcal{F} ist das Maximum von $\mathcal{N}(\mathcal{F}, S, \gamma)$ über alle m -elementigen $S \subset X$.

Satz 17 Sei \mathcal{F} eine Menge von Funktionen $f : X \rightarrow \mathbb{R}$ und $\gamma > 0$. Für jede Wahrscheinlichkeitsverteilung D auf $X \times \{-1, 1\}$ gilt für jede Hypothese $f \in \mathcal{F}$ mit Margin $m_S(f) \geq \gamma$ auf einer gemäß D gezogenen Beispiele-Menge S mit $|S| > 2/\varepsilon$ mit Wahrscheinlichkeit $1 - \delta$, dass der Klassifikationsfehler $F_D(f)$ kleiner gleich

$$\frac{2}{|S|} \left(\log \mathcal{N}(\mathcal{F}, 2|S|, \gamma/2) + \log \frac{2}{\delta} \right)$$

ist.

Beweis Analog zum Beweis von Satz 16. □

Analog zu Lemma 1 gilt folgendes Lemma, mit dem wir die Brücke zwischen der Überdeckungszahl und der fat-shattering-Dimension schlagen:

Lemma 2 *Sei \mathcal{F} eine Familie von Funktionen $X \rightarrow [a, b]$ und D eine Verteilung auf X . Sei $0 < \gamma < 1$ und $d := \text{fat}_{\mathcal{F}}(\gamma/4)$. Dann gilt:*

$$\log \mathcal{N}(\mathcal{F}, l, \gamma) \leq 1 + d \log \frac{2el(b-a)}{d\gamma} \log \frac{4l(b-a)^2}{\gamma^2}$$

Daraus erhalten wir:

Satz 18 \mathcal{F} bestehe aus Funktionen $X \rightarrow [0, 1]$ und es seien $\gamma > 0$ und eine Verteilung D auf $X \times \{-1, 1\}$ gegeben, nach der eine Stichprobe S mit mehr als $2/\varepsilon$ zusammengestellt sei. Sei f eine Hypothese mit Margin $m_S(f) \geq \gamma$ und es sei $\text{fat}_{\mathcal{F}}(\gamma/8) < |S|$. Dann ist mit Wahrscheinlichkeit $1 - \delta$ der Fehler von f kleiner als:

$$\frac{2}{l} \left(d \log \frac{8el}{d\gamma} \log \frac{32l}{\gamma^2} + \log \frac{4}{\delta} \right)$$

Um das auf SVM anzuwenden, benötigen wir die fat-shattering-Dimension linearer Funktionen-Klassen – oder zumindest eine Abschätzung:

Satz 19 Sei $R > 0$ und $X := \{x \in V : \|x\| \leq R\}$ und sei

$$\mathcal{L} = \{X \rightarrow \mathbb{R}, x \mapsto \langle w, x \rangle : \|w\| \leq 1\}$$

Dann gilt

$$\text{fat}_{\mathcal{L}}(\gamma) \leq \left(\frac{R}{\gamma}\right)^2$$