universitätbonn
Rheinische
Friedrich-Wilhelms-
Universität Bonn

Bachelorarbeit

Numerische Analyse des TrueSkill Verfahrens

Johannes Loevenich

Datum der Abgabe

Betreuung: Prof. Dr. Jochen Garcke

Fakultät für Mathematik
Rheinische Friedrich-Wilhelms- Universität Bonn

Inhaltsverzeichnis

Das Problem n Spieler zu bewerten ist ein wichtiger Forschungsbereich des Maschinellen Lernens.
Aus der Kombinatorik lässt sich schnell herleiten, dass es $n!$ verschiedene Möglichkeiten gibt n Spieler in einem Ranking zu bewerten. Ziel ist es das eine Ranking zu finden, welches das tatsächliche Können der einzelnen Spieler am besten wiederspiegelt. Nimmt man an, dass alle Rankings gleich wahrscheinlich sind, so würde dies bedeuten, dass $log_2(n!) \approx nlog_2(n)$ Spielausgänge nötig wären, um die korrekte Bewertung zu ermitteln. Diese Schranke ist jedoch nur dann aussagekräftig, falls die Ausgänge eines jeden Spiels gleichverteilt sind. Bei vielen Spielen wird durch das matchen von nahezu gleich starken Gegnern versucht diese Chancengleichheit zu erzwingen. In dieser Arbeit wird eine Wahrscheinlichkeitstheoretische Interpretation dieses Problems betrachtet, weshalb es im weiteren Verlauf von Bedeutung ist gewisse Unsicherheiten in Rang eines Spielers zu berücksichtigen. Interessanterweise reduziert sich die minimale Anzahl von aussagekräftigen Spielen auf $nlog_2(m)$, wenn wir annehmen, dass es $m \ll n$ Äquivalenzklassen oder Level gibt.
Betrachten wir Spiele bei denen k Teams gegeneinander antreten und bewertet werden, dann erfüllt jedes Spiel $log_2(k)$ Ausgänge und es werden nur $\frac{nlog_2(n)}{log_2(k!)}$ aussagekräftige Spielausgänge benötigt. Man wird sehen, dass das hier vorgestellte Verfahren für das betrachtete Problem nahezu optimal ist, also bis auf geringe Abweichungen gegen die oben genannten Schranken konvergiert.

1 Motivation

Skill Ratings erfüllen bei Online Spielen auf Konsolen (z.B. XBOX 360) und im Sport drei hauptsächliche Funktionen. Erstens lassen sich damit interessante und ausbalancierte, teambasierte Spiele zwischen Spielern mit ähnlichem Skill erzeugen. Zweitens ist es möglich Skills und Rankings für Spieler öffentlich zu machen, um damit zwischen den Spielern einen Wettbewerb zu erzeugen. Dirttens können Ratings als Qualifikationskriterium für Turniere verwendet werden. Mit dem steigenden Interesse an Online Spielen in den Letzten Jahren, ist auch das Interesse an effektiven Rating Systemen für Modelle mit Millionen Spielern pro Tag stark angestiegen.
1959 entwickelte Arpad Elo ein statistisches Rating System für Schach Spieler, welches von der *World Chess Federation FIDE* 1970 offiziell anerkannt wurde. Die grundlegende Idee des *Elo Rating Systems* ist es den möglichen Ausgang eines Spieles als Funktion der beiden Spieler Ratings s_1 und s_2 zu modellieren. In jedem Spiel weißt ein Spieler eine gewisse *Performanz* $p_i \sim \mathcal{N}(p_i; s_i, \beta^2)$ auf. Es wird angenommen, dass diese *Performanz* normalverteilt um s_i mit fester Varianz β^2 ist. Die Wahrscheinlichkeit, dass Spieler 1 gewinnt entspricht der Wahrscheinlichkeit, dass seine *Performanz* p_1 größer ist, als die *Performanz* p_2 seines Gegners

$$P(p_1 > p_2 | s_1.s_2) = \Phi(\frac{s_1 - s_2}{\sqrt{2}\beta}). \tag{1.1}$$

Φ bezeichnet die kummulative Dichte der Gaußverteilung mit mit Nullerwartung und

Einheitsvarianz. Nachdem das Spiel beendet ist werden die Skill Ratings der Spieler s_1 und s_2 so aktualisiert, dass der beobachtete Spielausgang wahrscheinlicher wird. Sei $y = +1$, falls Spieler 1 gewinnt, $y = -1$, falls Spieler 2 gewinnt und $y = 0$, falls die beiden Spieler unentschieden spielen. Das resultierende *Elo Update* für beide Spieler hat dann die Form $s_1 \leftarrow s_1 + y\Delta$ und

$$\Delta = \alpha\beta\sqrt{\pi}(\frac{y+1}{2}\Phi(\frac{s_1 - s_2}{\sqrt{2}\beta})). \tag{1.2}$$

Viele moderne Ranking Systeme basieren in ihrer Idee auf dem hier kurz vorgestellten *Elo Ranking System*. Mark Glickman zum Beispiel entwarf basierend auf dem *Elo Ranking System* das sogenannte *Glicko Rating System* um dem Problem entgegen zu wirken, dass das *Elo Ranking System* an die 20 Spiele benötigt um aussagekräftige Skill Ratings zu erzeugen.

Eine wichtiges neues Anwendungsgebiet für Ranking Systeme sind Online Multiplayer Spiele. Aufgabe ist es unter anderem ausgeglichene, faire Spiele zwischen Spielern mit ähnlichem Skill zu erzeugen. Multiplayer Online Spiele stellen folgende neue Anforderungen an Ranking Systeme:

1. Spielausgänge bestehen meist aus dem Ranking der Teams. Eine Herausforderung ist es, die Ranking Skills einzelner Spieler aus diesen Rankings zu erzeugen.

2. Es gibt Spiele bei denen mehr als zwei Teams oder Spielern gegeneinander antreten. Bei solchen Spielen gibt es nicht nur einen Gewinner und einen Verlierer. Viel mehr besteht der Spielausgang hier aus einer Permutation von Teams oder Spielern.

In dieser Arbeit werden wir ein Verfahren analysieren, dass diese Anforderungen unterstützt.

2 Das Ranking Problem

Wir betrachten im Folgenden Spiele, bei denen Spieler in zwei oder mehr Teams gegeneinander antreten. Sind nur zwei Spieler vorhanden, so betrachten wir die beiden Spieler als zwei Teams die gegeneinander antreten. Falls zwei Spieler oder Teams dieselbe Bewertung erhalten, sagen wir das das Spiel unentschieden ausgegangen ist. Im einfachen Fall von nur zwei gegeneinander antretenden Teams, sind nur drei Spielausgänge möglich: Sieg, Niederlage oder unentschieden.
Nummerieren wir alle teilnehmenden Spieler eines Spiels von 1 bis n, so kann ein Spiel zwischen k Teams vollständig durch die k Indizes $i_j \in \{1, ..., n\}$ der n_j Spieler im j-ten Team beschrieben werden. Der von jedem Team erreichte Rang sei dann definiert als $\mathbf{r} := (r_1, ..., r_k)^T \in \{1, ...k\}^k$. Es wird davon ausgegangen, dass dass der Gewinner eines Spiels den Rang 1 erhält.
Gesucht wird nach dem Skill s_i eines jeden Spielers, $\mathbf{s} \in \mathbb{R}^n$; wobei der Skill des j-ten Teams eine Funktion $S(s_{i_j})$ in Abhängigkeit aller Skills der Spieler des Teams ist. Im trivialen Fall, dass jedes Team nur einen Spieler enthält beschreibt $S(s_i) = s_i$ also die Identität. Für Skills wird die folgende Eigenschaft gefordert.

Definition 2.1. ***(Stochastische Transitivität)*** *Falls ein Team u vor einem Team v platziert wurde, so ist es wahrscheinlicher, dass Team u gegen Team v gewinnt als umgekehrt.*

$$S(s_{i_u}) \geq S(s_{i_v}) \Rightarrow P(\ \textit{Team u gewinnt}\ ) > P(\ \textit{Team v gewinnt}\ ) \tag{2.1}$$

3 Bayssche Methodik

Offensichtlich kennen wir den Skill eines Spielers nie mit absoluter Sicherheit. Deshalb basieren das Trueskill, sowie viele andere bekannte Verfahren auf Baysschen Modellen. Dazu beschreiben wir den Skill eines Spielers durch eine Wahrscheinlichkeitsverteilung $P(\mathbf{s})$. Wir nehmen außerdem an, dass diese Verteilung einer multivariaten Gaußverteilung entspricht, deren Kovarianzmatrix Diagonalmatrix mit Einträgen σ_i^2 ($P(\mathbf{s}) = \mathcal{N}(\mathbf{s}, \mu, \mathbf{diag}(\boldsymbol{\sigma^2}))$). Diese Annahme hat einige Vorteile:

1. Die Verteilung ist unimodal in $\mu \in \mathbb{R}^n$. Ein Spieler hat deshalb genau einen unbekannten Skill. Dieser schwankt nicht unerwartet zwischen verschiedenen Spielen.

2. Die Verteilung impliziert einfache Skillupdate Funktionen. (Siehe **TODO: Kapitel einfügen!**)

3. Die Verteilung kann Speichereffizient implementiert werden. Eine Diagonalmatrix kann leicht als Vektor abgespeichert werden. Der mittlere Skill μ_i und die Skillvarianz σ_i^2 sind für jeden Spieler Konstanten.

3.1 Spielausgangsbasierte Updates

Sei $\mathbf{r}$ der beobachtete Ausgang eines Spieles. Wir nehmen an, dass $\mathbf{r}$ als Vektor vorliegt, deren Einträge die Platzierungen der Spieler wiederspiegelen. So können wir die Skills aller Spieler $P(\mathbf{s})$ mithilfe der Baysschen Regel aktualisieren:

$$\begin{aligned} P(s|r, \{i_1, ..., i_k\}) &= \frac{P(r|s, \{i_1, ..., i_k\}P(s|\{i_1, ..., i_k\}))}{P(r|\{i_1, ..., i_k\})} \\ &= \frac{P(s|S(s, i_1), ..., S(s, i_k)\})P(s)}{P(r|\{i_1, ..., i_k\})} \end{aligned} \tag{3.1}$$

Diese neue Verteilung wird *Posterior* Verteilung genannt und wird im nachfolgenden Spiel für $P(s)$ benutzt. Diese Art der Vorgehensweise ist als *on-line learning* bekannt: Jederzeit existiert nur eine Verteilung $P(s)$, die durch jeden Spielausgang beeinflusst wird. Wir nehmen außerdem an, dass die *Posterior* Verteilung alle verfügbaren Informationen enthält. In der Praxis ist es schwer diese Größe exakt, kompakt und effizient darzustellen. Deshalb wird sie meist z.B. mithilfe von geeigneten Methoden approximiert. Wir stellen in dieser Arbeit zwei unterschiedliche Vorgehensweise dar um dieses Problem zu lösen. Ein auf Faktorgraphen und dem Expectation Propagation Algorithmus basierender Ansatz wird in Kapitel **TODO: Kapitel einfügen!** vorgestellt. In Kapitel **TODO: Kapitel einfügen!** wählen wir einen direkten Ansatz und versuchen auftretende Integrale numerisch möglichst effizient zu approximieren.

3.2 Zeitabhängige Updates

Die bis hierhin beschriebene Vorgehensweise macht eine durchaus starke Annahme:
Die Skills aller Spieler, μ, sind zeitunabhängig.
Dies ist in der Realität meist jedoch nicht der Fall. Denn Spieler können zum Beispiel mit der Zeit mehr über das Spiel lernen oder spielen eine Zeit lang nicht und werden wieder schlechter.
Wir sind deshalb an einem *Posterior* $P(s_i| \triangle t)$, welcher impliziert, dass ein Spieler mit Index i eine Zeit lang $\triangle t$ nicht gespielt hat. Dazu benötigen wir ein Maß $P(\triangle \mu| \triangle t)$ für die Änderung des Skills $\triangle \mu$ des Spielers in der Zeit $\triangle t$. Wir folgen dem Trueskill Modell nach Herbrich (Literatur) und nehmen an, dass der Skill eines Spielers in dieser Zeit steigt oder fällt und das Ausmaß der Änderung durch eine Funktion τ in Abhängigkeit von $\triangle t$ beschrieben wird. Damit erhalten wir $P(\triangle \mu| \triangle t) = \mathcal{N}(\triangle \mu; 0; \tau^2(\triangle t))$. Übertragen wir dieses Modell auf die Skills s_i der einzelnen Spieler so erhalten wir:

$$
\begin{aligned}
P(s_i|\triangle t) &= \int P(s_i|\mu_i + \triangle\mu) P(\triangle\mu|\triangle t) d(\triangle\mu) \\
&= \int \mathcal{N}(s_i; \mu_i + \triangle\mu, \sigma_i^2) \mathcal{N}(\triangle\mu; 0, \tau^2(\triangle t)) d(\triangle y) \\
&= \int \mathcal{N}(s_i; \mu) \mathcal{N}(\mu; \mu_i, \tau^2(\triangle t)) d\mu \\
&= \mathcal{N}(s_i; \mu_i, \sigma_i^2 + \tau^2(\triangle t))
\end{aligned}
\tag{3.2}
$$

Wir werden in dieser Arbeit τ als konstant τ_0 annehmen.

4 Wahrscheinlichkeitstheorie

Dieses Kapitel gibt einen Überblick über einige wichtige Konzepte der Wahrscheinlichkeitstheorie. Im Folgenden werden Mengen mit Großbuchstaben, z.B. X, und deren Elemente mit Kleinbuchstaben, z.B x, bezeichnet. Für Mengen definiere die Indikatorfunktion I_X durch

$$
I_X(x) := \begin{cases} 0 & \text{falls } x \notin X \\ 1 & \text{falls } x \in X \end{cases}
$$

Definition 4.1. ***(σ-Algebra)*** *Sei eine Menge χ gegeben. Ein Mengensystem Υ von Mengen $X \subseteq \chi$ wird genau dann σ-Algebra über χ genannt, wenn*

1. *Ist eine Menge $X \in \Upsilon$ enthalten, so auch ihr Komplement $X^c = \chi \backslash X$.*
2. *Falls $X_i \in \Upsilon, i = 1, ..., \infty$ abzählbares Mengensystem in Υ ,dann sind auch $\cup_{i=1}^{\infty} X_i \in \Upsilon$ und $\cap_{i=1}^{\infty} X_i \in \Upsilon$ in Υ enthalten.*

Kurz, jede σ-Algebra ist abgeschlossen unter Komplementbildung und abzählbaren Vereinigungen oder Schnitten.

Definition 4.2. ***(Borelmengen)*** *Sei $\chi = \mathbb{R}^n$, die Borelmengen B_n sind die kleinsten σ-Algebren, die alle offenen Intervalle*

$$
\{(x_1, ..., x_n) \in \mathbb{R}^n | \forall i \in \{1, ..., n\} : x_i \in (a_i, b_i)\}
$$

für alle $a_i, b_i \in \mathbb{R}$. Bemerke, dass B_n überabzählbar ist.

Definition 4.3. ***(Maß- und Wahrscheinlichkeitsraum)*** *Ein messbarer Raum ist ein Tupel (χ, Υ). Dabei ist χ das Universum und Υ σ-Algebra über χ. Ein Wahrscheinlichkeitsraum ist ein Tripel (χ, Υ, P), wobei P ein Wahrscheinlichkeitsmaß auf χ ist. D.h. $P : \Upsilon \to [0, 1]$, sodass $P(\chi) = 1$ und für disjunkte abzählbare Vereinigungen $X_i \in \Upsilon$, $i = 1, ...\infty$ gilt*

$$
P(\cup_{i=1}^{\infty} X_i) = \sum_{i=1}^{\infty} P(X_i).
$$

Definition 4.4. ***(Messbarkeit)*** *Sei (χ, Υ) messbarer Raum. Eine reellwertige Funktion $g : \chi \to \mathbb{R}$ heisst Υ-messbar (oder messbar) genau dann, wenn*

$$\forall z \in \mathbb{R} : \{x \in X | g(x) \leq z\} \in \Upsilon.$$

Definition 4.5. ***(Zufallsvariable)*** *Sei (χ, Υ) messbarer Raum. Eine Zufallsvariable ist eine χ-messbare reellwertige Funktion $f : \chi \to \mathbb{R}$.*

Eine Zufallsvariable $Y = f(X)$ induziert also ein Maß P_Y auf $\mathbb{R}$, für das die σ-Algebra A die Intervalle der Form $(-\infty, z)|z \in \mathbb{R}$ enthält. Das Maß P_Y ist vom Maß P_X und f indzuiert. Das bedeutet

$$\forall Y \in B_1 : P_Y(Y) := P_X(\{x \in X | f(x) \in Y\}).$$

Definition 4.6. ***(Verteilungsfunktion und Dichte)*** *Für eine Zufallsvariable X heißt die durch*

$$F_X(x) := P_X(X \leq x)$$

definierte Funktion $F_X : \mathbb{R} \to [0, 1]$ Verteilungsfunktion von X. Die Funktion $f_X : \mathbb{R} \to \mathbb{R}$ wird Dichte genannt, falls

$$\forall z \in \mathbb{R} : F_X(z) = \int_{x \leq z} f_X(x) dx.$$

Für weitere Betrachtungen ist Erwartung einer Zufallsvariablen von essentieller Bedeutung.

Definition 4.7. ***(Erwartungswert)*** *Sei $f : \chi \to \mathbb{R}$ messbare Funktion. Der Erwartungswert $E_X[f(X)]$ von f über die Wahrscheinlichkeit von x wird durch*

$$E_X[f(X)] := \int_{\mathbb{R}} f(x) dF_X(x)$$

definiert. Der Erwartungswert ist nur dann definiert, wenn $\int_{\mathbb{R}} |f(x)| dF_X(x) < \infty$.

Definition 4.8. ***(Varianz)*** *Die Varianz $Var(X)$ einer Zufallsvariable X ist definiert durch*

$$Var(X) := E_X[(X - \mu)^2] = E_X[X^2] - \mu^2,$$

wobei $\mu = E_X[X]$ der Erwartungswert der Zufallsvariable X ist.

Definition 4.9. ***(Produktraum)*** *Seien zwei Maßräume (χ, Υ) und (φ, Φ) gegeben. Definiere den Produktraum durch $(\chi \times \varphi, \Upsilon \times \Phi)$. Hierbei bezeichnet $\Upsilon \times \Phi$ die kleinste σ-Algebra welche die Mengen $\{X \times Y | X \in \Upsilon, Y \in \Phi\}$. enthält.*

Definition 4.10. ***(Marginal und bedingte Wahrscheinlichkeiten)*** *Sei $(\chi \times \varphi, \Upsilon \times \Phi, P_{XY})$ der Produktraum von X und Y. Das Marginalwahrscheinlichkeitsmaß P_X ist dann durch*

$$\forall X \in \Upsilon : P_X(X) := P_{XY}(X \times \varphi)$$

definiert. Sei $Y \in \Phi$ *und* $P_Y(Y) > 0$*, dann ist das bedingte Wahrscheinlichkeitsmaß* $P_{X|Y \in Y}$ *durch*

$$\forall Y \in \Upsilon : P_{X|Y}(X) := \frac{P_{XY}(X \times Y)}{P_Y(Y)}$$

gegeben.

Definition 4.11. ***Unabhängigkeit*** *Zwei Zufallsvariablen* X *und* Y *werden genau dann unabhängig genannt, wenn*

$$\forall X \in \Upsilon : \forall Y \in \Phi : P_{XY}(X \times Y) = P_X(X) P_Y(Y).$$

In diesem Fall genügen die Marginalverteilungen um den gesamten Produktraum zu definieren.

Im Folgenden schreibe $\mathbf{X}$, falls $\mathbf{X}$ eine Folge $(X_1, ..., Xn)$ von Zufallsvariablen definiert. Eine solche Folge kann je nach Kontext entweder als Zeilen- oder als Spaltenvektor interpretiert werden. Ein Element des Universums χ^n wird dann durch ein n-Tupel $\mathbf{x}$ beschrieben. Sei $\mathbf{x} = (x_1, ..., x_n)$ ein solches n-Tupel, dann verstehe $x \in \mathbf{x}$ als $\exists i \in \{1, ..., n\} : x_i = x$.

Definition 4.12. ***(Erwartungswert einer n-dimensionalen Zufallsvariable)*** *Seien* $\boldsymbol{X} = (X_1, ..., X_n)$ *n Zufallsvariablen mit gemeinsamem Wahrscheinlichkeitsmaß* P_X*, dann ist die Erwartung* $E_{\boldsymbol{X}}[\boldsymbol{X}]$ *durch das n-Tupel*

$$E_{\boldsymbol{X}}[\boldsymbol{X}] = (E_{X_1}[X_1], ..., E_{X_n}[X_n])$$

gegeben.

Definition 4.13. ***(Kovarianz und Konvarianzmatrix)*** *Seien* X *und* Y *zwei Zufallsvariablen mit gemeinsamem Wahrscheinlichkeitsmaß* P_{XY}*, dann ist die Kovarianz* $Cov(X, Y)$ *durch*

$$Cov(X, Y) := E_{XY}[(X - \mu)(Y - \nu)]$$

definiert, wobei $\mu = E_X[X]$ *und* $\nu = E_Y[Y]$*. Es ist leicht einzusehemn, dass* $Cov(X, X) = Var(X)$ *Sei* $\boldsymbol{X} = (X_1, ..., X_n)$ *eine Folge von n Zufallsvariablen und* $\boldsymbol{Y} = (Y_1, ..., Y_n)$ *eine weitere solche Folge mit gemeinsamer Dichte* P_{XY}*, dann ist die* $n \times m$ *Kovarianzmatrix* $\boldsymbol{Cov(X, Y)}$ *definiert durch*

$$\boldsymbol{Cov(X, Y)} := \begin{pmatrix} Cov(X_1, Y_1) & ... & Cov(X_1, Y_m) \\ . & ... & . \\ . & ... & . \\ . & ... & . \\ Cov(X_n, Y_1) & ... & Cov(X_n, Y_m) \end{pmatrix}$$

Falls $\boldsymbol{X = Y}$ $\boldsymbol{Cov(X,X) = Cov(X)}$

4.1 Eigenschaften von Zufallsvariablen

Satz 4.14. ***(Erwartungswert von Summen und Produkten)*** *Seien X und Y zwei unabhängige Zufallsvariablen, so gilt*

$$E_{XY}[X \cdot Y] = E_X[X] \cdot E_Y[Y], \tag{4.1}$$

$$E_{XY}[X + Y] = E_X[X] + E_Y[Y], \tag{4.2}$$

immer dann, wenn die Terme auf der rechten Seite existieren. Die zweite Aussage gilt sogar, falls X und Y nicht unabhängig sind.

Satz 4.15. ***(Linearität des Erwartungswertes)*** *Für jede n-dimensionale Zufallsvariable $\boldsymbol{X}$, jede Matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ und jeden stationären Vektor $\boldsymbol{b} \in \mathbb{R}^m$ gilt*

$$E_X[\boldsymbol{A}\boldsymbol{X} + \boldsymbol{b}] = \boldsymbol{A}E_X[\boldsymbol{X} + \boldsymbol{b}].$$

Satz 4.16. ***Varianz Zerlegung*** *Seien X und Y zwei unabhängige Zufallsvariablen, dann gilt*

$$Var(X + Y) = Var(X) + Var(Y)$$

Beweis 1. *Setze $\mu = E_X[X]$ und $\nu = E_Y[Y]$. Benutze die Definition der Varianz*

$$\begin{aligned} E_{XY}[(X + Y - E_{XY}[X + Y])^2] &= E_{XY}[((X - \mu)(Y - \nu))^2] \\ &= E_{XY}[(X - \mu)^2 + 2 \cdot (X - \mu)(Y - \nu) + (Y - \nu)^2] \\ &= E_X[(X - \mu)^2] + 2E_{XY}[(X - \mu)(Y - \nu)] + E_Y[(Y - \nu)^2] \\ &= E_X[(X - \mu)^2] + 2E_X[(X - \mu)]E_Y[(Y - \nu)] + E_Y[(Y - \nu)^2] \\ &= Var(X) + Var(Y) \end{aligned}$$

Lemma 4.17. *Für jede Zufallsvariable X und jede Konstant $c \in \mathbb{R}$ gilt $Var(cX) = c^2 \cdot Var(X)$.*

4.2 Mehrdimensionale Gauß Verteilung

In diesem Abschnitt werden einige der wichtigsten Eigenschaften der Gaussverteilung zusammengefasst.

Definition 4.18. *Sei $\mu \in \mathbb{R}^n$ ein Vektor und $A \in \mathbb{R}^{nxm}$ eine deterministische Matrix. Sei weiter $Y = (Y_1, ..., Y_m)$ eine Folge von m unabhängigen, normalverteilten Zufallsvariablen Y_i mit Mittewert Null und Einheitsvarianz ($Y_i \sim Normal(0, 1)$). Dann wird $X = AY + \mu$ normal- oder gaußverteilt mit Erwartungswert $E_X[X] = \mu$ und Kovarianzmatrix $Cov(X) = \Sigma = AA'$ genannt. Da das Maß P_X eindeutig durch diese beiden Größen beschrieben wird schreiben wir im Weiteren auch $Y \sim Normal(\mu, \Sigma)$.*

Satz 4.19. *Falls $X \sim Normal(\mu, \Sigma)$, dann besitzt X genau dann eine Dichte f_X, wenn Σ positiv definit ist. Die Dichte f_X ist gegeben durch*

$$f_X(x) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)) \tag{4.3}$$

Satz 4.20. *Sei $X \sim Normal(\mu, \Sigma)$ eine n-dimensional normalverteilte Zufallsvariable, $A \in R^{mxn}$ stationäre Matrix und $b \in \mathbb{R}^m$ stationärer Vektor. Dann ist auch die Zufallsvariable $Y = AX + b$ normalverteilt mit $Y \sim Normal(A\mu + b, A\Sigma A^T)$.*

Satz 4.21. *Angenommen $P_{X|Y=y} = Normal(Xy, \Gamma)$ ist normalverteiltes Wahrscheinlichkeitsmaß, wobei $X \in \mathbb{R}^{mxn}$ und $\Gamma \in \mathbb{R}^{mxm}$ stationäre Matrizen für alle Werte $y \in \mathbb{R}^n$ sind. Falls $P_Y = Normal(\mu, \Sigma)$ normalverteiltes Wahrscheinlichkeitsmaß ist, so gilt*

$$P_{Y|X=x} = Normal(\Psi(X^T\Gamma^{-1}x + \Sigma^{-1}\mu), \Psi). \tag{4.4}$$

$$P_X = Normal(X\mu, \Gamma + X\Sigma X^T). \tag{4.5}$$

,wobei $\Psi = (X^T\Gamma^{-1}X + \Sigma^{-1})^{-1}$

Beweis 2. *Nach Satz () wissen wir, dass*

$$f_{Y|X=x(y)} = \frac{f_{X|Y=y}f_Y(y)}{\int_{\mathbb{R}^n} f_{X|Y=y'(x)}f_Y(y')dy'} = \frac{f_{X|Y=y}(x)f_Y(y)}{f_X(x)}.$$

Es fällt auf, dass der Nenner unabhängig von y ist. Betrachte nun deshalb zunächst den Zähler. Mithilfe von Definition ist Letzteres gegeben durch

$$c \cdot exp(-\frac{1}{2}((x - Xy)^T\Gamma^{-1}(x - Xy) + (y - \mu)^T\Sigma^{-1}(y - \mu))),$$

wobei $c = (2\pi)^{\frac{-m+n}{2}}|\Gamma|^{-\frac{1}{2}}|\Gamma|^{-\frac{1}{2}}$ unabhängig von x und y ist. Diesen Ausdruck wiederum können wir umschreiben als

$$c \cdot exp(-\frac{1}{2}((y - x)^T C(y - c) + d(x))),$$

mit

$$C = X^T\Gamma^{-1}X + \Sigma^{-1},$$

$$Cc = X^T\Gamma^{-1}X + \Sigma^{-1}\mu,$$

$$d(x) = (x - X\mu)^T(\Gamma + X\Sigma X^T)^{-1}(x - X\mu).$$

Da $d(x)$ als Funktion nicht von y abhängt, kann der Term $exp(-\frac{1}{2}d(x))$ mit in die Konstante c gezogen werden und somit folgt die erste Gleichung des Satzes indem $\Psi := C^{-1}$ gesetzt wird.
Um die Zweite Aussage zu zeigen kann die Definition von $f_X(x)$ benutzt werden, d.h,

$$\begin{aligned} f_X(x) &= \int_{\mathbb{R}^n} c \cdot exp(-\tfrac{1}{2}((y' - c)^T C(y' - c) + d(x))dy' \\ &= c \cdot exp(-\tfrac{1}{2}d(x)) \cdot \int_{\mathbb{R}^n} exp(-\tfrac{1}{2}((y' - c)^T C(y' - c))dy' \\ &= c \cdot exp(-\tfrac{1}{2}d(x)) \cdot (2\pi)^{\frac{n}{2}}|C|^{\frac{1}{2}} = c' \cdot exp(-\tfrac{1}{2}d(x)) \\ &= c' \cdot exp(-\tfrac{1}{2}(x - X\mu)^T(\Gamma + X\Sigma X^T)^{-1}(x - X\mu)), \end{aligned}$$

wobei die dritte Zeile aus der Definition und der Tatsache folgt, dass sich Wahrscheinlichkeitsdichten zu Einer zusammenfassen lassen. Dies zeigt die zweite Aussage.

5 Faktorgraphen

Ein Faktorgraph ist ein bipartiter Graph, der beschreibt wie eine globale Funktion in Abhängigkeit von verschiedenen Variablen und Faktoren durch ein Produkt von lokalen Funktionen ausgedrückt werden kann.

5.1 Allgemeines

In Faktorgraphen wird zwischen zwei Typen von Knoten unterschieden: Die einen Knoten, welche mit Variablen identifiziert werden (nicht ausgefüllte Knoten) und die Anderen, welche mit lokalen Funktionen identifiziert werden (ausgefüllte Knoten). Kanten verbinden Variablenknoten x_i und Funtionenknoten f genau dann, wenn x_i Argument von f ist.

Sei $X = \{x_i\}_{i \in \mathbb{N}}$ eine Menge von Variablen bzgl. der Indexmenge $N = \{1, 2, 3, ..., n\}$. Falls E triviale Teilmenge von N ist, so bezeichne mit X_E die Teilmenge von X, welche durch E induziert wird. Für jedes $i \in N$ nehme die Variable x_i Werte aus dem Alphabet A_i an. Desweiteren wird angenommen,dass A_i für alle $i \in N$ stets endlich ist. Bezeichne eine bestimmte Belegung der Variablen aus X als Konfiguration der Variablen. Diese Konfigurationen können als Kartesisches Produkt $W = \prod_{i \in N} A_i$, den sogenannten Konfigurationsraum verstanden werden.

Ein Element $w = (w_1, ..., w_n) \in W$, mit $w_i \in A_i$ ist entspricht der Varaiblenbelegung $x_1 = w_1, ..., x_n = w_n$.

Im weiteren sind Funktionen mit Urbild W von besonderem Interesse. Sei $g : W \to R$ eine solche Funktion, auch gloabele Funktion genannt. Im Moment beziehe sich der Wertebereich auf die Reellen Zahlen. Im Allgemeinen sei jedoch jeder beliebige Semiring erlaubt.

5.2 Beispiel

Beispielsweise jeder lineare Binärblockcode kann durch eine Menge von Paritätstestgleichungen beschrieben werden, sodass jede Gleichung eine Bedingung an das Codewort $x = (x_1, ..., x_n)$ beschreibt, z.B, $\sum_{i \in E} x_i = 0$. Der zu einem Binärcode mit der angegebenen Paritätsmatrix

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \tag{5.1}$$

korrespondierende Faktorgraph wird in Figur () gezeigt.

5.3 Indikatorfunktion und Posterior Wahrscheinlichkeit

Anlehnend an das vorherige Beispiel, sei $(x_1, ..., x_n)$ ein gleichverteilt gezogenes Codewort, dass über einen Speicherlosen Kanal übermittelt werden soll und $y = (y_1, ..., y_n)$ die gesamte Ausgabe des Kanals. Die gemeinsame *a posterior* Wahrscheinlichkeit vin $\{x_1, ..., x_n\}$ kann dann durch eine Funktion f ausgedrückt werden

$$f(x_1, ..., x_n) = \prod_{E \in Q} f_E(x_E) \prod_{i=1}^{n} f(y_i|x_i). \tag{5.2}$$

Dabei ist für jeden Wert von x_i, $f(y_i|x_i)$ die korrespondierende Likelyhood Funktion am entsprechenden Ausgang y_i des Kanals. Die Faktorgraphdarstellung dieses Szenarios ist in Figur (b) dargestellt.

5.4 Baysschе Netwerke

Baysschе Nezwerke sind gerichtete, azyklische graphische Modelle auf einer Menge von Zufallsvariablen. Jeder Knoten v eines Baysschen Netwerkes wird mit einer Zufallsvariable assoziert. Bezeichnet $a(v)$, die Menge der Vorgänger des Knotens v, so hat die Wahrscheitsverteilung des Baysschen Netzwerkes die Form

$$p(v_1, v_2, ..., v_n) = \prod_{i=1}^{n} p(v_i|a(v_i)) \tag{5.3}$$

Falls $a(v_i) = \varnothing$ (d.h. v_i hat keine Vorgänger), so setze $p(v_i|\varnothing) = p(v_i)$. Figur **FIGUR EINFÜGEN** zeigt beispielsweise eine Bayssches Netwerk, welches die Faktorisierung

$$p(v_1, v_2, v_3, v_4, v_5) = p(v_1|v_2)p(v_2)p(v_3|v_2, v_4)p(v_4)p(v_5|p_4)$$

beschreibt.

5.5 Sum-Product Algorithmus

Sei $g(X)$ globale Funktion über die Variablen der Menge $X = \{x_i : i \in N\}$, wobei Variable x_i Werte der endlichen Menge A_i annimmt. In diesem Abschnitt wird eine Algorithmus beschrieben, um die marginal Funktionen

$$G_i(x_i) = \sum_{x_1 \in A_1, ..., x_{i-1} \in A_{i-1}, x_{i+1} \in A_{i+1}, ..., x_n \in A_n} g(x_1, ..., x_n) \tag{5.4}$$

für Variablen $x_i, i \in N$ zu berechnen. Im Weiteren sei $\sum_{x_i} f(x_i) = \sum_{x_i \in A_i} f(x_i)$ und genauso sei für eine Teilmenge $J \subset N$ mit $\sum_{x_i; i \in J}$ f(X) die Summe über alle möglichen Konfigurationen der Variablen x_i über J gemeint. Damit gilt $G_i(x_i) = \sum_{x_j; j \in N \setminus \{i\}} g(X)$. Die Definition der marginalen Funktion G kann nun auf eine Teilmenge J von N ausgeweitet werden.

$$G_i(x_i) = \sum_{x_i; i \in N \ J} g(X). \tag{5.5}$$

Falls $g(x_1, ..., x_n)$ eine Wahrscheinlichkeitsverteilung beschreibt, so ist $G_i(x_i)$ die Marginalverteilung und $G_J(X_J)$ die gemeinsame Wahrscheinlichkeitsverteilung der Variablen über die Indexmenge J.
Ist die Anzahl n der Argumente von g klein, so nutze eine alternative Kurzschreibweise für die Marginalfunktionen. Schreibe statt einem Argument x_i von g ein + um anzudeuten, dass über diese Variable summiert wird.

5.6 Funktionsweise des Algorithmus

Der Sum-product Algorithmus operiert mithilfe einer "message passing"Porzedur, die Produukte von lokalen Funktionen entlang der Pfade des Faktorgraphen aufsammelt. Es wird angenommen,dass dieser Graph ein Baum ist, d.h. dass dieser Graph keine Kreise enthält. Die Beschreibung des Algorithmus kann durch die Annahme vereinfacht werden, dass jeder Knoten wie ein Prozessor Nachrichten über Kanten übermittelt und empfängt.

Für diese vereinfachte Betrachtung arbeitet der Algorithmus wie folgt. Die Basisoperation an jedem Knoten ermittlet das Produkt aller eingehenden Nachrichten an diesem Knoten. Für Knoten, die eine Menge von Variablen darstellen, wird dieses Produkt um die zugehörigen lokalen Funktionen erweitert. Die so ermittelten Produkte werden dann mit dem Vorbehalt, dass Nachrichten über ausgehende Knoten keine Faktoren enthalten, über die ausgehenden Kanten übermittelt. Da vorrausgesetzt wurde, dass der Faktorgraph keine Kreise enthält, enthält das Produkt der ausgehenden Nachrichten einer Kante und der empfangenen Nachrichten dieser Kante alle Faktoren der globalen Funktion.

Diese sogenannte "message-passing"Prozedur wird von den Blättern des Faktorgraphs aus gestartet und iteriert über alle Knoten des Graphen. An Blättern, die Variablenmengen darstellen entspricht die ausgehende Nachricht einer Representation der lokalen Funktion dieses Knotens. An Blättern, die eine Variable darstellen entspricht sie hingegen der Indikatorfunktion. Alle anderen Knoten des Graphen warten zunächst bis sie genügend Nachrichten gesammelt haben um eine ausgehende Nachricht zu produzieren. Genauer soll das heißen, sie warten solange, bis an jeder bis auf einer eingehenden Kante Nachrichten empfangen wurden. Tritt dieser Fall ein, so wird das Produkt aller eingehenden Nachrichten mit der lokalen Funktion gebildet und über die freie Kante übermittelt. Wird an dieser übrig gebliebenen Kante eine Nachricht empfangen, so werden die Produkte der zugehörigen lokalen Funktion über alle anderen ausgehnden Kanten versendet. Diese Prozedur wird in Figur **FIGUR EINFÜGEN!!!** illustriert.

Dieser Algorithmus wird dann effektiv, wenn man beachtet, dass von lokalen Funktionen über Pfade gesammelte Produkte marginalisiert werden können. D.h., dass nicht

alle Variablen entlang einer Kante beachtet werden müssen. Im Allgemeinen muss eine Variable beachtet werden, wenn sie Argument einer nachfolgenden lokalen Funktion ist. Andernfalls kann sie vernachlässigt werden.

Figur **FIGUR EINFÜGEN!!!** zeigt das Fragment eines Faktorgraphen. Die Update Regeln für dieses Fragment ergeben sich dann also

$$\mu_{x\to A}(x) = \mu_{B\to x}(x) \cdot \mu_{C\to x}(x) \tag{5.6}$$

$$\mu_{A\to x}(x) = \sum_{y,z} f_A(x,y,z) \cdot \mu_{y\to A}(x) \cdot \mu_{z\to A}(x) \tag{5.7}$$

$$F_x(x) = \mu_{x\to A}(x) \cdot \mu_{A\to x}(x) \tag{5.8}$$

5.7 Belief Propagation in Baysschen Netzwerken

FIGUR EINFÜGEN!!!
Die Verteilungsfunktion () eines Bayssschen Netzwerkes erlaubt eine intuitive Umformulierung zur Representation eines Faktorgraphen. Eine zu einem einzigen Faktor gehörende lokale Funktion in (), hat die Form $f(x|a(x))$, wobei $a(x)$ die Menge der Nachfolger von x im zugehörigen Baysschen Netzwerk ist. In Faktorgraphen wurden Nachfolgerknoten durch Pfeile gekennzeichnet. Diese Pfeile erlauben es uns einen Faktorgraphen ebenso als Bayssches Netzwerk ansehen zu können.

6 Expection Propagation

In diesem Kapitel werden rekursive Approximationstechniken beschrieben, um die KL-Divergenz zwischen Posterior und Approximation zu minimieren. Das sogenannte Ässumed-density Filteringïst eine schnelle Methode auf diesem Gebiet. Ëxpectation Propagationöder kurz EP ist eine Erweiterung des Ässumed-density Filteringüm Situationen stapelweise zu verarbeiten. Es hat höhere Genauigkeit als das Ässumed-density Filteringünd andere vergleichbare Methoden um Schlussfolgerungen zu approximieren.

6.1 Assumed-density Filtering

Dieser Abschnitt fasst die Idee des Ässumed-density Filtering"(ADF) zusammen um die Grundlagen für die Methode des Ëxpectation Propagationßu schaffen. Ässumed-density Filteringïst eine der grundlegenden Techniken, um Posterior in Baysschen Netzwerken und anderen statistischen Modellen zu approximieren.

7 Ein Ansatz mittels Faktorgraphen für das Trueskill Verfahren

Das Trueskill Verfahren ist ein Bayssches Ranking Verfahren, dass in seinen Grundzügen eine Erweiterung des Elo Ranking Systems aus dem Schach ist. Dieses neue System beachtet Unsicherheiten in der Bewertung eines Spielers, stellt neue Modelle für ausgeglichene Spiele auf und kann die Skills eines einzelnen Spielers aus dem Rang eines Teams ermitteln. Unsicherheiten werden durch eine Approximierung des message passing in faktorgraphen simuliert.

7.1 Der Trueskill Faktorgraph

Sei $1, ..., n$ eine Menge von Spielern, die in k Teams gegeneinander antreten. Die Zuweisung der Teams sei durch k disjunkte Teilmengen $A_j \subset \{1, ..., n\}$ mit $A_i \cap A_j = \emptyset$ für $i \neq j$ eindeutig beschrieben. Der Ausgang des Spiels sei definiert durch $\mathbf{r} := (r_1, ..., r_k) \in \{1, ..., k\}$, wobei r_j der Rang eines jeden Teams j sei. Falls $r_j = 1$, so hat Team j das Spiel gewonnen. Falls $r_i = r_j$ für $i \neq j$ so sagen wir, dass Team i gegen Team j unentschieden gespielt hat.

Wir wollen die Wahrscheinlichkeit $P(r|s, A)$ eines bestimmten Spielausganges $\mathbf{r}$ mit gegebenen Skills $\mathbf{s}$ der Spieler und einer Teamzuweisung $A := (A_1, ..., A_k)$ modellieren. Mithilfe der Bayschen Regel erhalten wir für die Posterior Verteilung

$$p(\mathbf{s}|\mathbf{r}, A) = \frac{P(\mathbf{r}|\mathbf{s}, A)p(\mathbf{s})}{P(\mathbf{r}|A)}. \tag{7.1}$$

Dabei nehmen wir an, dass $p(\mathbf{s}) := \prod_{i=1}^{n} \mathcal{N}(s_i; \mu_i, \sigma_i^2)$. Dies macht Sinn, da der Faktorgraph selbst eine Menge von Multiplikationen darstellt, um eine gemeinsame Verteilung zu finden. Wir weisen außerdem jedem Spieler in einem Spiel eine Performanz $p_i \sim \mathcal{N}(p_i; s_i, \beta^2)$ mit Zentrum s_i und Varianz β^2 zu. Die Performanz t_j eines Teams j entspricht der Summe $t_j := \sum_{i \in A_j} p_i$ der Spieler des Teams.

Ordnen wir die Teams nach ihren Rängen im Spielausgang $\mathbf{r}$ und nehmen wir an, dass unentschieden zunächst nicht zugelassen sind, so erhalten wir

$$P(\mathbf{r}|\{t_1, ..., t_k\}) = P(\mathbf{r}|\{t_r(1), ..., t_r(k)\}). \tag{7.2}$$

Sind unentschieden zugelassen, so erfordert der Spielausgang $r_(j) < r_(j+1)$ $t_r(1) > t_r(j+1)+\epsilon$ und ein unentschieden der Art $r_(j) = r_(j+1)$ umgekehrt $|t_r(1)-t_r(j+1)| \leq \epsilon$, wobei $\epsilon > 0$ ein Maß für die Gewinnspanne ist.

Wir wollen nun nach jedem Spiel die Skills der Spieler in Abhängigkeit des Spielausganges $\mathbf{r}$ auktualisieren und nutzen dafür die Darstellung mittels Faktorgraphen und den daran gebundenen Expectation Propagation Algorithmus. Dazu approximieren wir die Posterior Verteilungen gaußverteilt und verwenden sie im darauf folgenden Spiel als Prior Verteilungen.

Um das Verfahren an einem Beispielgraphen herleiten und erklären zu können, betrachten wir folgendes Spiel. Wir nehmen an, dass $k = 3$ Teams mit $A_1 = \{1\}, A_2 =$

$\{2,3\}$ und $A_3 = \{4\}$ gegeneinander antreten. Der Spielausgang sei $\mathbf{r} = (1,2,2)$. Dies bedeutet, dass Team 1 das Spiel gewonnen hat und Team 2 und Team 3 unentschieden gespielt haben. In Figur () geben wir den korrespondierenden Trueskill Faktor Graphen an.

Wie bereits in Kapitel () beschrieben, ist die grundlegende Idee eine gemeinsame Wahrscheinlichkeitsverteilung von mehreren Zufallsvariablen in zwei verschiedene Typen von Knoten aufzuteilen. Schwarze Boxen stellen Faktorknoten und Kreise Variablenknoten dar. Die gemeinsame Verteilung ergibt sich dann als Produkt der einzelnen Faktoren:

$$p(X) = \frac{1}{Z} \prod_S f_S(X_S) \tag{7.3}$$

X_S beschreibt die zu einem Faktor f_S zugehörigen Variablen und Z sei Normalisierungskonstante. Das wichtigste Konzept ist dann die Anwendung der Summen-Produkt-Update Regel:

Jede Nachricht, die von einem Knoten v über eine Kante e übermittelt wird, entspricht dem Produkt der lokalen Funktion an v (Oder der Einheitsfunktion, falls v Variablenknoten ist) mit allen eingegangenen Nachrichten an v außer an Kante e.

In Anlehnung an Kapitel () ergeben sich damit die drei folgenden Gleichungen

$$p(v_k) = \prod_{f \in F_{v_k}} m_{f \to v_k}(v_k) \tag{7.4}$$

Gleichung () sagt uns, dass der Wert des Marginales an v_k dem Produkt der eingehenden Nachrichten entspricht.

$$m_{f \to v_j}(v_j) = \int ... \int f(v) \prod_{i \neq j} m_{v_i \to f}(v_i) d\mathbf{v}_{\setminus j} \tag{7.5}$$

Gleichung () zeigt, dass der Wert einer Nachricht von einem Faktor f zu einer Variable v_j der Summe über das Produkt aller anderen Nachrichten zwischen diesem Faktor und abhängigen Variablen außer v_j entspricht.

$$m_{v_k \to f}(v_k) = \prod_{f' \in F_{v_k} \setminus \{f\}} m_{f' \to v_k}(v_k) \tag{7.6}$$

Gleichung () beschreibt, dass die Nachricht von einer Variable v_k zu einem Faktor f das Produkt aller anderen Nachrichten ist.

Für weitere Details möchten wir auf Kapitel () und die dortigen Bespiele verweisen.

7.2 Prior Faktoren

Betrachten wir die erste Ebene des Trueskill Graphen Prior Faktor ist im Trueskill Graphen als schwarze Box dargestellt.

Um die Trueskill Update Formeln vereinfacht darstellen zu können, verwenden wir die folgenden Notationen

$$\begin{aligned} \pi &= \sigma^{-2} = \frac{1}{\sigma^2} \\ \tau &= \pi\mu = \frac{\mu}{\sigma^2} \end{aligned} \tag{7.7}$$

Das Ziel des Prior Faktors ist es diese Werte einer gaußverteilten Zufallsvariable mit Erwartung m und Varianz $v\nu^2$ zu aktualisieren. Mit obiger Notation ergibt sich also:

$$\begin{aligned} \pi_x^{\text{new}} &\leftarrow \pi_x + \frac{1}{\nu^2} \\ \tau_x^{\text{new}} &\leftarrow \tau_x + \frac{m}{\nu^2} \end{aligned} \tag{7.8}$$

Wir sehen, dass die beiden Gleichungen den ersten Gleichungen im Trueskill Paper () entsprechen.

7.3 Unsicherheiten

Wie bereits in der Einleitung erwähnt können Skills nur mit einer gewissen Unsicherheit beschrieben werden. Wir beschreiben hier, wie solche Unsicherheiten im Trueskill Verfahren berücksichtigt werden. Dazu nehmen wir an, dass wir eine gauqverteilte Zufallsvariable y gegeben haben und eine neue gaußerverteilte Zufallsvariable $\mathcal{N}(x; y, c^2)$ erhalten wollen, die eine gewisse Unsicherheit c^2 berücksichtigt. Dazu sei $a := (1+c^2(\pi_y - \pi_{f\to y}))^{-1}$. Durch Umformungen erhalten wir

$$\begin{aligned} a = (1 + c^2(\pi_y - \pi_{f\to y}))^{-1} &= \frac{1}{1 + c^2(\pi_y - \pi_{f\to y})} \\ &\approx \frac{\sigma_y^2}{\sigma_y^2 + c^2} \end{aligned} \tag{7.9}$$

Es ist somit leicht zu sehen, dass a stets kleiner als 1 ist. Nun kann die Ungewissheit c^2 approximativ den Werten π und τ zugewiesen werden:

$$\begin{aligned} \pi_{f\to x}^{\text{new}} &\leftarrow a(\pi_y - \pi_{f\to y}) = \frac{1}{\sigma_y^2 + c^2} \\ \tau_{f\to x}^{\text{new}} &\leftarrow a(\tau_y - \tau_{f\to y}) = \frac{\mu_y}{\sigma_y^2 + c^2} \end{aligned} \tag{7.10}$$

7.4 Teamperformanz

Dieser Faktor wird benutzt um die Differezen zwischen Teams innerhalb eines Spieles auszudrücken und summiert mehrere gaußverteilte Variablen miteinander. Wir nehmen

an, dass wir n Variablen $v_1, ..., v_n$ miteinander summieren wollen, wobei jede dieser Variablen mit einem Faktor a_n gewichtet wird und v_0 dem Wert der Summe entspricht. Es gilt also:

$$v_0 = a_1 v_1 + ... + a_n v_n \tag{7.11}$$

Aus Kapitel () über die Gaußverteilung wissen wir, dass dies einer Zufallsvariable mit Erwartungswert und Varianz der Form

$$\begin{aligned} \mu &= \sum_{i=1}^{n} a_i \mu_i \\ \sigma^2 &= \sum_{i=1}^{n} a_i^2 \sigma_i^2 \end{aligned} \tag{7.12}$$

entpricht. Da die Update Formeln durch den Gebrauch der oben eingeführten Notation mithilfe von π und τ einfacher und kompakter zu implementieren sind, müssen diese in eine solche Form gebracht werden.

8 Hochdimensionale Integration

In diesem Kapitel stellen wir verschiedene Integrationstechniken zur Approximation Hochdimensionaler Integrale

$$\int_\Omega f(\mathbf{x}) d\mathbf{x} \approx \mathcal{Q}f = \sum_{i=1}^{N} \omega_i f(x_{(i)}) \tag{8.1}$$

vor. Wir bezeichnen Ω als das d-dimensionale Integrationsgebiet und

$$\epsilon(f) := |Qf - \int_\Omega f(\mathbf{x}) d\mathbf{x}|. \tag{8.2}$$

8.1 Monte Carlo

Bei der Monte Carlo Integration wählen wir die Stützstellen $\S_{()}$ als Zufallszahlen einer d-dimensionalen Gleichverteilung und feste Gewichte $\omega_i = \frac{1}{N}$. Das so entstandene Verfahren

$$\int_\Omega f(\mathbf{x}) d\mathbf{x} \approx \frac{1}{N} \sum_{i=1}^{N} f(x_{(i)}) \text{ mit } x_{(i)} \sim \mathcal{U}_{\lceil}(\Omega) \tag{8.3}$$

hat nach dem Gesetz der Großen Zahlen für beschränkte Varianzen eine Konvergenzrate von

$$\epsilon(f) = \mathcal{O}(N^{-\frac{1}{2}}). \tag{8.4}$$

Wir sehen schnell, dass der Nachteil dieses Verfahrens in seiner geringen Konvergenzrate für niedrige Dimensionen liegt. Vorteile sind die geringen Vorraussetzungen an den Integranden und die Dimensionsunabhängigkeit der Konvergenzrate.

8.2 Quasi-Monte Carlo

Bei Quasi-Monte Carlo Verfahren verwenden wir statt Pseudo Zufallszahlen als Auswertungspunkte deterministische Punktfolgen so. Nieder-Diskrepanz-Folgen, die das Integrationsgebiet Ω überdecken. In dieser Arbeit haben wir Sobolev-, Niederreiter- und Ritchmeyer-Folgen () implementiert. Diese Änderung führt dazu, dass die Integrationspunkte gleichmäßiger auf dem Integrationsgebiet verteilt sind. Dies wird in Abbildung () deutlich.

Die verbesserten Verteilungseigenschaften liefern im geringen und mittleren Dimensionsbereich eine Verbesserung der Konvergenzrate auffassen

$$\mathcal{O}(\frac{\log(N)^d}{N}). \tag{8.5}$$

8.3 Produktansatz

Der Produktansatz basiert auf eindimensionalen Quadraturverfahren. Ein eindimensionales Quadraturverfahren ist durch

$$Q_l^{(1)} f := \sum_{i=1}^{N_l} \omega_i f(x_{li}) \tag{8.6}$$

mit Diskretisierungslevel l definiert.
Durch Tensorierung von eindimensionalen numerischen Quadraturformeln können wir eine Quadraturformel für hochdimensionale Integrale konstruieren. Sei l_i das Diskretisierungslevel der Quadraturformel in der i-ten Dimension und N_{li} die zugehörige Anzahl von Stützstellen, so lässt sich das Tensorprodukt der Quadraturformeln mit den i_l-ten Stützstellen und den i_l-ten Gewichten durch

$$\begin{aligned} Q^{(d)} f &= (Q_{l_1}^{(1)} \otimes ... \otimes Q_{l_d}^{(1)}) f \\ &= \sum_{i_1=1}^{N_{l_1}} ... \sum_{i_d=1}^{N_{l_d}} \omega_{l_1} \cdot ... \omega_{l_d} \cdot f(x_{l_1 i_1}, ..., x_{l_d i_d}) \end{aligned} \tag{8.7}$$

definieren.

Ein Nachteil dieses Ansatzes ist die exponentiell wachsende Anzahl von Auswertungspunkten (Fluch der Dimension): Ein Quadraturverfahren mit der gleichen Diskretisierung in jede Richtung, also mit $N := N_{l_1} = ... = N_{l_d}$-Auswertungspunkten, benötigt in

d Dimensionen ingesamt N^d Auswertungspunkte.

Wir erhalten als Konvergenzrate des Verfahrens

$$\epsilon(f) = (O)(n^{-\frac{\alpha}{d}}) \tag{8.8}$$

Hierbei beschreibt α die Konvergenzrate des eindimensionalen Quadraturverfahrens. Aufgrund der exponentiell von der Dimension d abhängenden Konvergenzrate ist dieses Verfahren für hohe Dimensionen unbrauchbar.

8.4 Eindimensionale Quadraturverfahren

Wir beschränken uns in dieser Arbeit zunächst auf die Clenshaw-Curtis Regel aus (). Neben den Gauß-Quadraturformeln ist diese eine der bedeutensten eindimensionalen Quadraturregeln und für die in Kapitel () auftretenden Integrale gut geeignet.

8.4.1 Clenshaw-Curtis

Auch wenn die Clenshaw-Curtis Regel keinen optimalen Exaktheitsgrad hat, ist sie aufgrund ihrer geschachtelten Struktur der Gewichte und Stützstellen für die Integrations mittels dünner Gitter sehr hilfreich.
Die Stützstellen und Gewichte der offenen Clenshaw-Curtis-Formeln ergeben sich als Extrempunkte der Chebycheff-Polynome auf $(0, 1)$

$$\begin{aligned} x_{l_i} &= \frac{1}{2}(1 - \cos(\frac{\pi i}{N_l + 1})) \\ w_{l_i} &= \frac{2}{N_l + 1} \sin(\frac{\pi i}{N_l + 1}) \sum_{j=1}^{\frac{(N_l+1)}{2}} \frac{1}{2j-1} \sin(\frac{(2j-1)\pi i}{N_l + 1}). \end{aligned} \tag{8.9}$$

Der Index l gibt das Level und $N_l := 2^l - 1$ die Anzahl der Stützstellen bzw. der Gewichte an.

8.5 Dünne Gitter

Ein Ansatz dem eben erwähnten Fluch der Dimension zu entgehen bieten die sogenannten Dünnen Gitter. Damit kann erreicht werden, dass die Konvergenzrate, ähnlich zu Quasi-Monte Carlo Techniken, nur logarithmisch von der Dimension des Problems abhängt.

Wir wählen den Ansatz mittels Differenz von zwei Quadraturformeln unterschiedlicher Level. Sei eine Folge von Quadraturformeln mit $N_l^{(d)}$ Auswertungspunkten für ein Level $l \in \mathbb{N}$ mit $N_l^{(d)} < N_{l+1}^{(d)}$ gegeben.
Wir definieren die Differenz zwischen zwei Leveln durch

$$\begin{aligned}\Delta_k^{(1)} f &= (Q_k^{(1)} - Q_{k-1}^{(1)}) f \\ &= \sum_{i=1}^{N_k} \omega_{k,i} f(x_{k,i}) - \sum_{i=1}^{N_k} \omega_{k-1,i} f(x_{k-1,i}) \end{aligned} \tag{8.10}$$

und $Q_0 f := 0$.

Mittels dieser Differenzen lässt sich die klassische Dünngitter-Quadratur nach Smolyak (siehe) mit $k \in \mathbb{N}^d$ als

$$Q_l^{(d)} f = \sum_{k_1+...+k_d \leq l+d-1} (\Delta_{k_1}^{(1)} \otimes ... \otimes \Delta_{k_d}^{(1)}) f \tag{8.11}$$

definieren. Anders als beim Produktansatz werden in der Dünngitter-Quadratur nur die Summanden betrachtet, deren Summe der Level echt kleiner als die vorgebene Konstante $l + d$ ist.

Für geschachtelte Quadraturformeln, wie die vorgestellte Clenshaw-Curtis-Regel, lässt sich die Teleskopsumme in () zu

$$\sum_{i=1}^{N_k} \omega_{k,i} f(x_{k,i}) \text{ mit } \omega_{k,i} = \begin{cases} \omega_{k,i} & \text{falls i ungerade} \\ \omega_{k,i} - \omega_{k-1,\frac{i}{2}} & \text{falls i gerade} \end{cases} \tag{8.12}$$

vereinfachen.

Zur Fehleranalyse dieses Verfahrens betrachten wir die Klasse $\mathcal{W}_d^r$ der Funktionen mit beschränkten Ableitungen bis zur Ordnung r:

$$\mathcal{W}_d^r := \{g : \Omega \to \mathbb{R}, ||\frac{\delta^{|s|} g}{\delta^{|s_1|} x_1, ..., \delta^{|s_d|} x_d}||_\infty < \infty, s_i \leq r\}. \tag{8.13}$$

Erfüllt die eindimensionale Quadraturregel zusätzlich

$$\epsilon(f) = \mathcal{O}(n_l^{-r}) \tag{8.14}$$

und gilt $N_l = \mathcal{O}(2^l)$, dann ergibt sich für alle Funktionen $d \in \mathcal{W}_d^r$ die Fehlerabschätzung

$$\epsilon(f) = \mathcal{O}(N_l^{-r} (\log N_l)^{(d-1)(r+1)}) \tag{8.15}$$

für die Integration mittels dünner Gitter. Der Unterschied der dünnen Gitter für die Clenshaw-Curtis-Regel gegenüber dem Produktansatz ist anschaulich in Abbildung () zu erkennen.

Literatur

Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbständig verfasst und keine anderen, als die angegebenen Quellen und Hilfsmittel benutzt, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht und die Satzung des Karlsruher Instituts für Technologie zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet habe.

Ort, den Datum