Beweise $\mathcal{NP}$-Vollständigkeit

Für Entscheidungsproblem $L \overset{?}{\in} \mathcal{NP}$-vollständig:

1. Zeige, dass $L \in \mathcal{NP}$: Zertifikat, Größe, Verifikator, Polynomialzeit
2. Zeige, dass L $\mathcal{NP}$-hart ist mit $L' \leq_p L$, wo L' ein bekanntes $\mathcal{NP}$-vollständiges Problem ist. Definiere eine Abbildung f, die jede Instanz $x \in L'$ in eine Instanz $f(x) \in L$ transformiert:
   (a) Beschreibe **Gadgets**: die Einschränkungen des zu zeigenden Problems.
   (b) Zeige, dass $f(x)$ in **Polynomialzeit** erfolgt.
3. Korrektheit: Zeige $x \in L' \iff f(x) \in L$:
   (a) $\Rightarrow$: Wenn x eine Ja-Instanz von L' ist, dann besitzt $f(x)$ eine Lösung in L.
   (b) $\Leftarrow$: Zeige, dass die Gadgets nur eine dem ursprünglichen Problem entsprechende Lösung zulassen.

Beziehung $NPO \sim \mathcal{NP}$

Ein Optimierungsproblem $P = (I, S, \mu, \text{opt})$ liegt in NPO, wenn die Instanzmenge I entscheidbar ist, die Lösungen $S(x)$ polynomiell beschränkt sind und die Zielfunktion μ effizient berechenbar ist.

Schritt 1: Definiere das Ents.problem P'

Benutze für P' einen gegebenen Schwellenwert B:

$$P' = \left\{ (x, B) \mid x \in I,\ \exists y \in S(x) \quad \text{mit } \mu(x, y) \geq B \text{ (bzw. } \leq B) \right\}$$

Schritt 2: Zertifikat

- **Zertifikat:** Als Zertifikat z für eine Instanz x wähle eine potenzielle Lösung $y \in S(x)$.
- **Polynomielle Größe:** Da $P \in NPO$, existiert ein Polynom p, sodass für alle $y \in S(x)$ gilt: $|y| \leq p(|x|)$. Das Zertifikat ist somit polynomiell beschränkt.

Schritt 3: Verifikator $M(x, B, z)$

Muss die folgenden in Polynomialzeit ausführen:

1. **Instanzprüfung:** Teste, ob $x \in I$ (möglich da $I \in P$) oder ob x polynomiell kodierbar ist.
2. **Maßberechnung:** Berechne $v = \mu(x, z)$ unter Verwendung der Maschine M_μ (möglich da μ in Polynomialzeit berechenbar ist).
3. **Schwellenwert-Vergleich:** Prüfe, ob v das Kriterium $\text{opt}(v, B)$ erfüllt (z. B. $v \geq B$ bei Maximierung).

Chernoff-Schranken

Seien $X_1, \ldots, X_n$ nun n unabhängige binäre Zufallsvariablen mit $\text{Ws}(X_i = 1) = p_i$. Sei $X = \sum_{i=1}^{n} X_i$ eine reelle Zufallsvariable, dann gilt für $\delta > 0$:

$$\begin{aligned} \text{Ws}[X \geq (1+\delta)\,\mathbb{E}(X)] &\leq \left[\frac{e^{\delta}}{(1+\delta)^{1+\delta}} \right]^{\mathbb{E}(X)} \\ &\leq \exp\left(-\frac{\mathbb{E}(X)\,\delta^2}{3} \right) \end{aligned}$$

$$\begin{aligned} \text{Ws}[X \leq (1-\delta)\,\mathbb{E}(X)] &\leq \left[\frac{e^{-\delta}}{(1-\delta)^{1-\delta}} \right]^{\mathbb{E}(X)} \\ &\leq \exp\left(-\frac{\mathbb{E}(X)\,\delta^2}{2} \right) \end{aligned}$$

Ungleichung von Chebyshev

Seien $X_1, \ldots, X_n$ n unabhängige binäre Zufallsvariablen mit $\text{Ws}(X_i = 1) = p_i$. Sei $X = \sum_{i=1}^{n} X_i$ eine reelle Zufallsvariable, dann gilt für $\delta > 0$:

$$\text{Ws}(|X - \mathbb{E}(X)| \geq \delta) \leq \frac{V(X)}{\delta^2}$$

Entwurf von Approx.algorithmen

Sei Problem P gegeben mit Eingabe (Instanzen), Lösungen und Optimum (μ).

a) Zeige $P \in \mathcal{NPO}$

1. Eingabe: Die Instanzen müssen polynomiell überprüfbar sein.
2. Lösung: muss polynomiell in der Eingabegröße beschränkt sein.
3. Maß (μ) muss in Polynomialzeit berechenbar sein.

b.1) Algorithmus (r-Approx)

r ist gegeben.

- Benutze einen Greedy-Ansatz, um für jedes Element lokal optimal zu entscheiden. Das heißt, z.B., ein Element wird entschieden/zugewiesen, so dass es an dem Moment den geringsten Schaden oder höchsten Nutzen erzeugt für die Maßfunktion.
- Zeige, dass die Laufzeit polynomiell ist.

b.2) Korrektheit / Approx.güte Γ_A

Zeige: das Verhältnis zwischen dem Optimum und dem Algorithmusergebnis (y) ist beschränkt. Hier wird Maximierung erläutert. „Elementanzahl“ steht informell für die Elemente, die in die Maßberechnung fließen.

- **Untere Schranke des Algorithmus:** Zeige, welchen Wert der Algorithmus mindestens erzielt. $\rightarrow$ Oft $\frac{\text{Elementenzahl}}{r}$ für Maximierung.
- **Verhältnis zum Optimum:** Setze dies in Relation zum Optimalen. Optimum ist ja $\leq$ Elementanzahl für Maximierung.
- **Güte berechnen:**

$$\Gamma_A(x,y) \leq \frac{\text{Elementanzahl}}{\frac{\text{Elementenzahl}}{r}} = r$$

PTAS-Reduktion

Übertrage die Approximierbarkeit eines Problems A auf ein Problem B. Hier müssen nicht nur die Instanzen, sondern auch die Fehlerparameter und Lösungen abgebildet werden.

Reduktion

Eine PTAS-Reduktion $A \leq_{PTAS} B$ besteht aus einem Tripel $\rho = (f, g, \alpha)$ mit folgenden Eigenschaften:

1. Beschreibe die Reduktionsidee. $\rightarrow$ $f(x, \epsilon)$: Überführt eine Instanz x von A und einen Fehlerparameter ϵ in eine Instanz von B.
2. Beschreibe, wie die Lösung von B in die Lösung von A zurücktransformiert wird: $\rightarrow g(x, y, \epsilon)$: Überführt eine Lösung y der Instanz $f(x, \epsilon)$ zurück in eine Lösung für x im Problem A.
3. Definiere α (oft $\alpha(\epsilon) := \epsilon$). Berechnet aus dem gewünschten Fehler ϵ für Problem A den notwendigen Fehler für Problem B.
4. Zeige, dass f,g und α in Polynomialzeit berechenbar sind.

Korrektheit

5. Zeige, dass die Lösung von B mithilfe von $g()$ zu einer zulässigen Lösung für A abgebildet wird. Benutze hierfür die beschriebene $f - g$-Beziehung.

Approximationsgüte

6. Zeige:

$$\Gamma_B(f(x,\epsilon), y) \leq 1+\alpha(\epsilon) \implies \Gamma_A(x, g(x,y,\epsilon)) \leq 1+\epsilon$$

   (a) Wissen: Güte für eine Lösung von B $\leq 1 + \alpha(\epsilon) = 1 + \epsilon$

   (b) Benutze $g(x, y, \epsilon)^*$ um zu zeigen: $\frac{\text{A-Lösung}}{\text{A-Optimum}} \overset{*}{=} \frac{\text{B-Lösung}}{\text{B-Optimum}} \overset{\text{wie oben}}{=} 1 + \epsilon$

Eigenschaften einer Metrik

Eine Abbildung $d : X \times X \to \mathbb{R}$ heißt Metrik auf einer Menge X, wenn für alle $x, y, z \in X$ die folgenden drei Axiome erfüllt sind:

- **Definitheit (Positive Definitheit):** $d(x, y) \geq 0$ und $d(x, y) = 0$ genau dann, wenn $x = y$.
- **Symmetrie:** $d(x, y) = d(y, x)$.
- **Dreiecksungleichung:** $d(x, z) \leq d(x, y) + d(y, z)$.

Sum-of-Pairs Cost

Ab hier ist $\tilde{w} : \bar{\Sigma}_0^2 \to \mathbb{R}$ eine Kostenfunktion.

$$w(a_1, \ldots, a_k) = \sum_{i=1}^{k} \sum_{j=i+1}^{k} \tilde{w}(a_i, a_j)$$

Center-Star Cost

$$w(a_1, \ldots, a_k) = \sum_{\{i,j\} \in E} \tilde{w}(a_i, a_j)$$

Konsensus-Kostenfunktion

$$w'(a_1, \ldots, a_k) = \min \left\{ \sum_{i=1}^{k} w(a_i, c) : c \in \bar{\Sigma} \right\}$$

Konsensus-Fehler

$$E_S(s') := \sum_{j=1}^{k} d(s', s_j)$$

Ein optimaler Steiner-String ist

$$s^* = \operatorname*{argmin}_{s' \in \Sigma^*} E_s(s')$$

Carrillo-Lipman Algorithm

1. Berechne P und S Matrizen.
2. Bestimme P+S Matrix durch zellenweise Addition.
3. Der Heap funktioniert wie folgt: Jede Zelle im Heap mit $\leq C$ soll alle nächsten (rechts, unten und unten diagonale) mit ins Heap nehmen. Am Anfang steht die Zelle links oben im Heap.
4. $C_{s,t} := C - \sum_{(s_i,s_j) \neq (s,t)} d(s_i, s_j) =^* C$ ($=^*$ gilt, wenn es nur zwei Sequenzen s, t gibt.)

C-optimale Schnittpositionsfamilie

Gegeben: eine Menge von Sequenzen S (hier drei: s_1, s_2, s_3) und ein Index c_1.
Gefragt: C-optimale Schnittpositionen und MSA.

1. Für paarweise verschiedene Sequenzen berechne die Präfix (P) und Suffix (S) Matrizen. Achte darauf, dass die S-Matrix ab der unten rechten Ecke ausgefüllt wird.
2. Berechne die Zusatzkostenmatrix (C): Zellenweise Addition von P und S minus der Alignment-Score.
3. Berechne die finale Matrix $C(c_1, i, j)$ mit

   $$C(c_1, i, j) := C_{s_1,s_2}(c_1, i) + C_{s_1,s_3}(c_1, j) + C_{s_2,s_3}(i, j)$$

   Das heißt: Halte in den Zusatzkostenmatrizen von s_1-s_2 und s_1-s_3 die c_1-te Zeile fest. Dann addiere die Zahl in der i-ten Spalte von der s_1-s_2-Matrix und die Zahl in der j-ten Spalte von der s_1-s_3-Matrix zu der Zelle (i, j) in der Matrix von s_2-s_3.
4. Bestimme in der finalen Matrix die kleinsten Zahlen und ihre Koordinaten. $\to$ C-optimale Schnittpositionen bzgl. $c_1 \to \{(c_1, x_1, y_2), (c_1, x_2, y_2), \ldots\}$
5. Koordinaten zu Alignments interpretieren: (c_1, x_1, y_2) $\to$ „Nimm die ersten c_1 Buchstaben von s_1, die ersten x_1 von s_2 und die ersten y_2 von s_3. Aligniere diese und die Restlichen optimal.“

Center-Star-Methode

1. Berechne paarweise Distanzen.
2. Bestimme den Center-String s_c.
3. Aligniere alle übrigen Sequenzen optimal gegen s_c.
4. Führe die paarweisen Alignments konsistent über den Center-String zu einem MSA zusammen.

Geliftete PMSA

Berechne rekursiv die Distanzen eines optimalen gelifteten PMSA für den Teilbaum T_v gewurzelt am Knoten v markiert mit der Sequenz s:

$$D(v,s) = \sum_{(v,w)\in E(T)} \min_{s'\in S(w)} \{d(s,s') + D(w,s')\}$$

$D(v,s) = \infty$ für $s \notin S(v)$ und $D(v,s) = 0$ für Blätter.

Mit **legalen Kantenpaaren**:

$$D(v,s) = \sum_{(v,w)\in E(T)} \min_{(s,s')\in L(v,w)} \{d(s,s') + D(w,s')\}$$

$D(v,s) = 0$ falls v Blatt mit $s_v = s$

Uniform: Ein gelifteter Baum heißt uniform, wenn für jeden Knoten eines Levels entweder alle gelifteten Sequenzen nur vom linken oder nur vom rechten Kind stammen. Durch eine Markierung der Wurzel mit $s \in S$ wird ein eindeutiges uniformes Lifting beschrieben.

PAM-Matrix

1. Relative Häufigkeiten:

$$p_a := \frac{1}{2n} \sum_{b\in\Sigma} n_{a,b}$$

2. Mutationswahrscheinlichkeiten ($a \neq b$):

$$p_{a,b} := \frac{n_{a,b}}{2n} \cdot \frac{1}{p_a} \cdot \frac{1}{100}$$

$$p_{a,a} := 1 - \sum_{\substack{b\in\Sigma \\ b\neq a}} p_{a,b}$$

3. als Kostenfunktion:

$$\begin{aligned} w(a,b) &= \log\left(\frac{p_a \cdot p_{a,b}}{p_a \cdot p_b}\right) \\ &= \log\left(\frac{n_{a,b}}{200 \cdot n \cdot p_a \cdot p_b}\right) \end{aligned}$$

$$\begin{aligned} w(a,a) &= \log\left(\frac{p_a \cdot p_{a,a}}{p_a \cdot p_a}\right) \\ &= \log\left(\frac{200 \cdot n \cdot p_a - \sum_{b\in\Sigma} n_{a,b}}{200 \cdot n \cdot p_a^2}\right). \end{aligned}$$

BLOSUM-Matrix

Für einen gegebenen r:

1. Teile die Sequenzen nach ihren Längen in Blocks auf.
2. In jedem Block gruppiere die Sequenzen in Clustern nach ihrer Ähnlichkeit, so dass die Sequenzidentität eines Clusters nach Single-Linkage nicht weniger als r% beträgt.
3. Bestimme $H^{(\beta)}_{p,q}(a,b)$: zähle Positionen in paarweisen Alignments, wo a und b gegenüberstehen, dividiert durch $|C_p| \cdot |C_q|$.
4. $H(a,b)$: Summiere die Matrizen zellenweise.
5. $$q_{ab} := \frac{H(a,b)}{\sum_{a,b\in\Sigma} H(a,b)}$$
$$p_a := \frac{\sum_{b\in\Sigma} H(a,b)}{\Sigma_{a,b\in\Sigma} H(a,b)}$$
6. $$w(a,b) := \log_2(\frac{q_{a,b}}{p_a \cdot p_b})$$

Maximum-Likelihood-Schätzer

1. Likelihood-Funktion: $L(p) := L(p; n) = \dots$
2. Ziel: Finde $\theta^* := \text{argmax}_{p\in\Theta}\{L([n \mid N, p])\}$
   Allgemein: $\text{argmax}\{\text{Ws}[X \mid \theta\,] : \theta \in \Theta]\}$
3. Umformen zu Log-Lik: $\ln(L(p)) = \dots$
4. Ableitung:
   $$\frac{\mathrm{d}}{\mathrm{d}p}\ln(L(p)) = \dots \overset{!}{=} 0 \to \text{Finde eine Nullstelle}$$
   .
5. Überprüfe ob Maximum mit
   $$\frac{\mathrm{d}^2}{\mathrm{d}p^2}\ln(L(p)) = \dots \overset{?}{<} 0$$
6. Behandle die Randfälle $p \in \{0, 1\}$ und erläutere, wie das für MLE kein Problem ist.

Maximum-A-Posteriori-Schätzer

$$\theta^*_{MAP} := \text{argmax}\,\{\text{Ws}[\theta \mid X]\} \tag{1}$$
$$(\text{bzw}) := \text{argmax}\,\{f(\theta \mid X) : \theta \in \Theta\} \tag{2}$$
$$:= \underset{\theta\in\Theta}{\text{argmax}}\,\{\log(f(x \mid \theta) + \log(f_0(\theta))\} \tag{3}$$

wo $\text{Ws}[\theta \mid X]$ die Posteriori-Wahrscheinlichkeit ist mit

$$\text{Ws}[\theta \mid X] = \frac{\text{Ws}[X \mid \theta] \cdot f_0[\theta]}{\text{Ws}[X]}$$

bzw. $f(\theta \mid x)$ eine Dichtefunktion für den Parameterraum Θ ist, nachdem die Daten gegeben sind (Posterior).

Hypothesentest

1. **Bestimme Richtung:** Was besagt die Alternative? Wie würde sie die Zufallsvariable beeinflussen?
2. **Ablehnungsbereich:** (Wsl von beobachteten und extremeren Ereignissen): $Ws(N) := \text{Ws}[X \le N]$ oder $\text{Ws}[X \ge N]$
3. **Signifikanz:** $Ws \overset{!}{=} \alpha$
4. Löse nach N.
5. Bestimme den (Nicht-)Ablehnungsbereich für die Werte von N.

Likelihood Ratio Test

1. $$\Lambda(x) = \frac{L(\theta_0; x)}{L(\theta_1; x)} = f(x)$$
2. **Significance Level (α):**
   $$P[\Lambda(x) \le \lambda \mid \theta_0] \overset{!}{=} \alpha$$
3. **Transformation to Critical Region:**
   $$\Lambda(x) \le \lambda \iff f(x) \lesseqgtr \lambda \iff x \lesseqgtr g(\lambda)$$
4. **Probability Distribution of X:**
   $$\begin{aligned} P[X \lesseqgtr g(\lambda) \mid \theta_0] &= P[X \lesseqgtr A] \\ &= (\text{use underlying prob distro}) \\ &= h(A). \end{aligned}$$
5. **Solve for Critical Value A:**
   $$h(A) = \alpha = 0.05$$
6. **Decision Rule:**
   $$\text{Entscheidung nach } X \lesseqgtr A$$

Markov-Kette

Zeige, dass eine Zufallsvariablenfolge Markov-Kette ist:

$$\text{Ws}[X_n = q_n \mid X_{n-1} = q_{n-1}] = \dots =$$
$$\text{Ws}[X_n = q_n \mid (X_{n-1}, \dots, X_1) = (q_{n-1}, \dots, q_1)]$$

Stationäre Verteilung $\hat{p}$ findet man mit:

$$\hat{p} \cdot P = \hat{p}$$

ergodisch $\iff$ irreduzibel $\wedge$ aperiodisch
Sei (Q, P, π) ein Markov-Modell.

1. **aperiodisch:** wenn alle Zustände $q \in Q$ aperiodisch sind (also Periode 1 haben). Die Periode d_q eines Zustands $q \in Q$ ist definiert als
   $$d_q := \text{ggT}\left\{k \in \mathbb{N} : \begin{array}{l} \exists(q_0, \dots, q_k) \in Q^{k+1} \wedge q_0 = q_k = q \\ \wedge\ \forall i \in [0 : k-1] p_{q_i, q_{i+1}} > 0 \end{array}\right\}.$$
2. **irreduzibel:** wenn es für alle Paare $(q, q') \in Q^2$ ein $k \in \mathbb{N}$ gibt, so dass $p^{(k)}_{q,q'} > 0$.