Stochastische Optimierungsalgorithmen

ADS: Algorithmen und Datenstrukturen 2

12. Teil

Peter F. Stadler & Konstantin Klemm

Bioinformatics Group, Dept. of Computer Science & Interdisciplinary Center for Bioinformatics, **University of Leipzig**

23. Juni 2010

Stochastische Optimierungsalgorithmen

Stochastische Optimierungsalgorithmen

Problemstellung

Lösungensmenge X, Bewertungsfunktion $f : X \to \mathbb{R}$.

Finde globales Minimum von f

Grundprinzip

- Start: endliche Menge A von erlaubten Lösungen ("Population") mit $|A| = n$.
- Schleife
  (1) Erzeuge erweiterte Lösungsmenge B, $|B| = m > n$.
  (2) Selektiere eine neue Population $A' \subset B$ mit $|A'| = n$ in der die besseren Lösungen angereichert sind.
- Abbruchbedingung

Stochastische Optimierungsalgorithmen

Stochastische Optimierungsalgorithmen

Motiviation durch Prozesse in der Natur:
Evolution als Wechselspiel von Mutation (allergemeiner: Erzeugung von Variabilität und Selektion (aus den vorhandenen Varianten)
Schrittweise Energie-Minimierung in physikalischen Systemen (z.B. Gefrieren von Flüssigkeiten ...)

Stochastische Optimierungsalgorithmen

Random Generate and Test

- $A = \{x\}$, eine einzelne Lösung
- $B = \{x, y\}$, wobei $y \in X$ eine **zufällig** gezogene Lösung ist.
- $A' = \{y\}$ if $f(y) < f(x)$, sonst $A' = \{x\}$.

Keine Garantie, dass ein globales Minimum gefunden wird.
FRAGE: wie kann man die *Struktur* von f ausnutzen?
IDEE: Gute Lösungen werden "irgendwie" ähnlich zu anderen guten Lösungen sein.

Stochastische Optimierungsalgorithmen

"Moves"

Lasse nur bestimmte "Schritte" zu.
Einfachster Fall: $A = \{x\}$
Formal: Definiere eine (erlaubte) Nachbarschaft $N(x)$ für jedes $x \in X$.
Nun waehle ein $y \in N(x)$ mit Wahrscheinlichkeit $1/|N(x)|$.
Impliziert eine Graph-Struktur Γ auf X: $(x, y) \in E$ iff $y \in N(x)$.
Notwendige Bedingung: Γ muss stark zusammenhängend sein sonst kann der Prozess auf einer Zusammenhangskomponente eingesperrt bleiben, die kein globales Minimum enthält.

Stochastische Optimierungsalgorithmen

Fitness-Landschaften

Formale Beschreibung des Optimierungsproblems als Funktion über der Knotenmenge des Graphen Γ
Die Frage nach geometrischen Eigenschaften wie

- Zahl von lokalen Minima
- Tiefe und Breite von Tälern
- Sattelpunkte zwischen Tälern
- Rauhheits-Maße

und deren Einfluss auf die Optimierung ist Gegenstand der Forschung
Veränderung des Move-Sets verändert die Landschaft und damit das Optimierungsverhalten.
Suche problemangepasste Movesets. Idee: einzene Moves sollten die Qualität der Lösung (meistens) nicht zu drastisch verändern.

Stochastische Optimierungsalgorithmen

Adaptive Walk

- **Start:** wähle $x \in X$ zufällig
- **Schleife:**
  (1) wähle $y \in N(x)$ gleichverteilt
  (2) falls $f(y) < f(x)$, setze $x \rightarrow y$.
- **Abbruchbedingung**

Folge der Lösungen enspricht einem Pfad in Γ, entlang dem f strikt abnimmt. ("Adaptive Walk")
Lokales Minimum: $f(y) \geq f(x)$ für alle $y \in N(x)$.
Pfade bleiben in lokalen Minima stecken. Bei geeigneter Abbruchbedingung erreichen alle Pfade ein lokales Minimum.
PROBLEM: Lösungen können sehr schlechter Qualität sein, d.h. $f(x) \gg f(\text{min})$.

Stochastische Optimierungsalgorithmen

Gradient Descent

- **Start:** wähle $x \in X$ zufällig
- **Schleife:**
  (1) bestimme die Menge $Z(x)$ aller $z \in N(x)$ für die $f(z)$ minimal über $N(x)$ ist.

  wähle $y \in Z(x)$ gleichverteilt
  (2) falls $f(y) < f(x)$, setze $x \rightarrow y$.
- **Abbruchbedingung**: stop falls $f(y) \geq f(x)$.

Folge der Lösungen enspricht einem Pfad in Γ, entlang dem f so steil wie möglich abnimmt. (Weg des steilsten Abstieg, "Gradient Descent".
Endet immer in einem lokalen Minimum.
Teurer als adaptive walk, weil $N(x)$ durchsucht werden muss.

Metropolis-Walks

Um lokalen Minima zu entkommen, lasse auch "aufwärts" Schritte zu:

- **Start:** wähle $x \in X$ zufällig
- **Schleife:**
  (1) wähle $y \in N(x)$ gleichverteilt
  (2) falls $f(y) < f(x)$, setze $x \longrightarrow y$

  sonst setze $x \longrightarrow y$ mit Wahrscheinlichkeit

  $$\exp(-(f(y) - f(x))/T)$$

- **Abbruchbedingung**

Steil bergauf = exponentiell selten
Eigenschaft: Wenn man **lange** genug wartet, besucht man zu einem bestimmten Zeitpunkt t die Lösung x mit Wahrscheinlichkeit

$$p(x) = \exp(-f(x)/T)/Z$$

Für kleine T: Lösungen mit kleinen Funktionswerten werden angereichert.

Stochastische Optimierungsalgorithmen

Simulated Annealing

IDEE: Reduziere die "Temperatur" T im Laufe der Simulation
Kirkpatrick, Gelatt and Vecchi (1983) and V. Černý (1985)
WICHTIGE ROLLE:
"cooling schedule" T_k, typischerweise eine monoton fallende Funktion der Schrittzahl k
Der Algorithmus konvergiert falls $T_k \geq d/\log(k)$ für große k.
Das ist in der Praxis aber zu langsam, man braucht exponentiell viele Schritte k um eine gegebene Genauigkeit $|f(x) - f(\min)|$ zu erreichen.
Anwendung auf harte Probleme, für die keine effizienten (polynomialen) Algorithmen bekannt sind.

Stochastische Optimierungsalgorithmen

Genetische Algorithmen

Cross-Over: "mischen" von 2 Eltern.
$\mathcal{R}(x, y)$ Menge der möglichen Kinder der Eltern x und y
Konstruktion der erweiterten Population:

1. Wähle eine Menge von Eltern-Paaren
2. Für jedes Elternpaar (x, y) ziehe einen, zwei oder mehrere Nachkommen $z \in \mathcal{R}(x)$.
3. Wähle eine Teilemenge von A und ziehe für jedes $x \in A$ einen Mutanten $y \in N(x)$.
4. Bilde die Vereinigung von A, den Cross-over Nachkommen, und den Mutanten

Stochastische Optimierungsalgorithmen

Genetische Algorithmen

Selektion:
viele Varianten:

- fitness-proportional selection.
  Ziehe aus B mit Wahrscheinlichkeiten proportional zu $-f(z)$ oder $\exp(-f(z))$ bis genau n Lösungen gezogen sind
- ziehe zwei Lösungen z und z'. Falls $z > z'$, akzeptiere z. Wiederhole bis n Lösungen akzeptiert sind.
- Wähle die besten n Lösungen.
- ...

Die Eingeweide: Kodierung

Explizite Representation der Lösungen

- Strings fixer Länge.
  Moves: Ersetzen eines Zeichens
  $N(x) = \{y | d_H(x, y) = 1\}$ d_H ... Hamming Distanz
- Permutationen.
  Moves: Transpositionen
- Reell-wertige Vektoren.
  Hier werden oft normalverteilte "Mutanten" gewählt:

$$p_{yx} \sim exp(-||x - y||^2/\sigma^2)$$

- auch andere kombinatorische Objekte können benutzt werden:
  Spannbäume, Partitionen, Matchings, ...

Stochastische Optimierungsalgorithmen

Cross-Over

Für Strings:

1-Punkt Crossover

```
x1 = 0100001010100010101010101101
x2 = 1010101110101010011100110101
```

Bestimme einen zufälligen *Bruchpunkt* k und vertausche die Suffixe:
$x_1' = x_1[1..k]x_2[k+1..n]$ und $x_2' = x_2[1..k]x_1[k+1..n]$
Für $k = 7$

```
x1' = 0100001.1101010100111100110101
x2' = 1010101.0101000101010101101101
```

Stochastische Optimierungsalgorithmen

Cross-Over

Uniformer Crossover

übernimm jede Position zufällig entweder unverändert oder vertausche die beiden Zeichen zwischen Eltern

```
    x1 = 01100011101010101011001001 01
    x2 = 10001010101000100110101111 01
tausch   --**-***--*******--**---**-*
```

Beobachtung: $d(x_1, x_1') \leq d(x_1, d_2)$ und $d(x_1, x_2') \leq d(x_1, d_2)$
Übliche Forderung an Rekombination:

- $\mathcal{R}(x, y) = \mathcal{R}(y, x)$
- Für $z \in \mathcal{R}(x, y)$ gilt $d(x, z), d(y, z) \leq d(x, y)$

Stochastische Optimierungsalgorithmen

Crossover für Vektoren

Mittelwertbildung:

$$z = \frac{1}{2}(x + y)$$

Konvexe Kombination:

$$z = px + (1 - p)y \qquad p \in [0, 1]$$

Die Zufallszahl p wird typischerweise aus einer Gleichverteilung gezogen
Spezielle Verfahren für andere Typen von Objekten, wie z.B., Permutationen, Bäume ...

Stochastische Optimierungsalgorithmen

Abbruchbedingungen

- Normalerweise ist nicht bekannt, ob das Ziel bereits erreicht ist, d.h., ob ein globales Minimum bereits gefunden ist.
- Fixe Anzahl von Iterationen
- Keine Verbesserung seit K Iterationen
- Verbesserung in den letzten K Iterationen weniger als ϵ (im Fall von reell-wertigen Problemstellungen)