

Bildverarbeitung: Diskrete Energieminimierung

Entrauschung → Segmentierung

Beide Definitionsbereich und Wertebereich sind diskret.

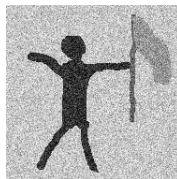
$R \in \mathbb{Z}^2$ – die Pixelmenge, $E \subset R^2$ – die Nachbarschaftstruktur (z.B. 4-Nachbarschaft)
 $x : R \rightarrow \mathbb{Z}$ – das Ausgangsbild, $y : R \rightarrow K$ – die gesuchte Abbildung (das restaurierte Bild). $k \in K$ repräsentiert den „wahren“ Grauwert (Label).

Die Energieminimierung:

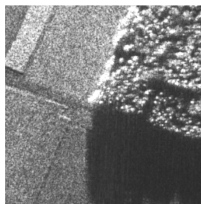
$$y^* = \arg \min_y [E_d(y) + \alpha E_m(y)]$$

$$\text{z.B. } \arg \min_y \left[\sum_{r \in R} (x_r - y_r)^2 + \alpha \sum_{rr' \in E} (y_r - y_{r'})^2 \right]$$

Farben → semantische Bedeutungen (Werte eines Merkmals)...



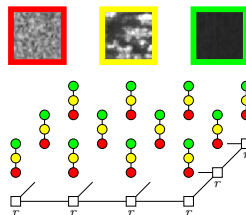
Die Menge der Pixel ist auf „sinnvolle“ Teilmengen zu partitionieren.



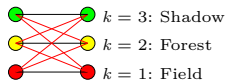
Original



A possible segmentation

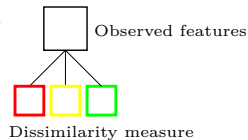


Compactness terms



— Penalty
— Zero

Data terms



$$y^* = \arg \min_y \left[\sum_r q_r(y_r) + \sum_{r,r'} g_{rr'}(y_r, y_{r'}) \right]$$

Iterated Conditional Modes

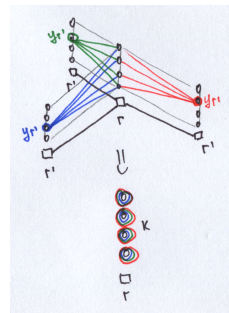
$$y^* = \arg \min_y \left[\sum_r q_r(y_r) + \sum_{r,r'} g_{rr'}(y_r, y_{r'}) \right]$$

Die Idee: wähle (lokal) immer wieder das energetisch günstigste Label bei fixiertem Rest [Besag, 1986].

Wiederhole oft für alle r :

$$y_r = \arg \min_k \left[q_r(k) + \sum_{r': rr' \in E} g_{rr'}(k, y_{r'}) \right]$$

(ME: synchrone Dynamik in Hopfield-Netzen)

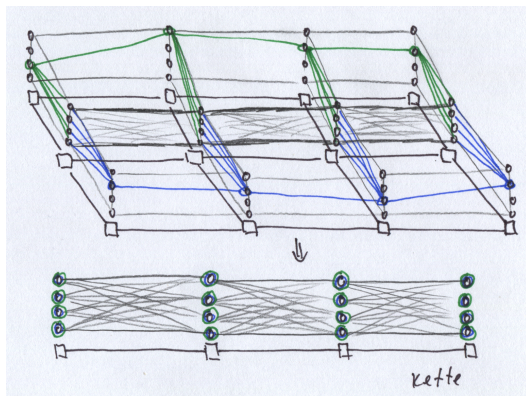


- + Extrem einfach, parallelisierbar.
- „Koordinatenweise“ Optimierung
 - konvergiert nicht zum globalen Optimum selbst bei einfachen Modellen.

Iterated Conditional Modes

Erweiterung: fixiere nicht alle Variablen bis auf eine, sondern nur eine Teilmenge so, das der Rest einfach optimierbar ist (zum Beispiel eine Kette oder ein Baum).

Für Bilder – Zeilenweise/Spaltenweise Optimierung.



→ durch Dynamische Programmierung exakt und effizient lösbar.

Äquivalente Transformationen (Reparametrisierung)

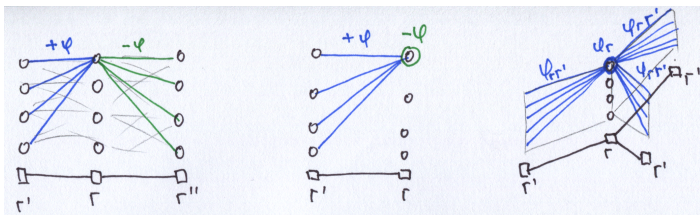
Zwei Aufgaben $A = (q, g)$ und $A' = (q', g')$ sind zu einander **äquivalent**, wenn

$$\left[\sum_r q_r(y_r) + \sum_{rr'} g_{rr'}(y_r, y_{r'}) \right] = \left[\sum_r q'_r(y_r) + \sum_{rr'} g'_{rr'}(y_r, y_{r'}) \right]$$

für alle Labellings y gilt.

$\mathcal{A}(A)$ – Äquivalenzklasse (alle zu A äquivalenten Aufgaben).

Äquivalente **Transformationen**:

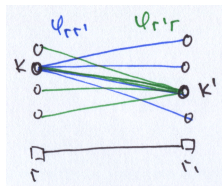


$$\Phi = (\varphi_r(k) \quad \forall r, k, \quad \varphi_{rr'}(k), \quad \forall rr', k)$$

$$\varphi_r(k) + \sum_{r': rr' \in E} \varphi_{rr'}(k) = 0 \quad \forall r, k$$

Äquivalente Transformationen

Sei $A = (q, g)$ eine Aufgabe, $A' = (q', g') = \Phi(A)$ ist die Aufgabe nach der Anwendung der Äquivalenten Transformation Φ , d.h.



$$q'_r(k) = q_r(k) + \varphi_r(k)$$

$$g'_{rr'}(k, k') = g_{rr'}(k, k') + \varphi_{rr'}(k) + \varphi_{r'r}(k')$$

$\Rightarrow A$ und A' sind zu einander äquivalent.

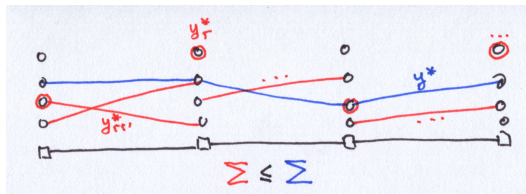
Sind zwei Aufgaben A und A' äquivalent,
so **existiert** eine Äquivalente Transformation Φ so, dass $A' = \Phi(A)$.

Weitere Eigenschaften:

$$\Phi(\Phi'(A)) = \Phi'(\Phi(A)) = (\Phi \oplus \Phi')(A) - \text{Superposition.}$$

$$\Phi^{-1}(\Phi(A)) = A, \text{ d.h. } \Phi \oplus \Phi^{-1} = \Phi^0 - \text{Inverse Transformationen.}$$

Die Menge aller Φ bildet eine **Gruppe**.



Die Energie einer Aufgabe:

$$E(A) = \min_y \left[\sum_r q_r(y_r) + \sum_{rr'} g_{rr'}(y_r, y_{r'}) \right]$$

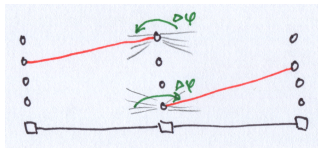
Scheinbare Qualität einer Aufgabe:

man wähle von einander unabhängig für jede Kante (für jeden Knoten) das beste Labelpaar (den besten Label)

$$SQ(A) = \sum_r \min_k q_r(k) + \sum_{rr'} \min_{kk'} g_{rr'}(k, k')$$

$SQ(A)$ ist offensichtlich eine **untere Schranke** für $E(A)$, d.h. $SQ(A) \leq E(A)$

Eine Aufgabe heißt **trivial**, wenn $E(A) = SQ(A)$ gilt.



Die Äquivalenten Transformationen ändern $E(A)$ **nicht**, $SQ(A)$ aber **schon**.

Die Idee – suche die Aufgabe größter Scheinbarer Qualität in der Äquivalenzklasse $\mathcal{A}(A)$
– maximiere die untere Schranke der Energie:

$$\sum_r \min_k (q_r(k) + \varphi_r(k)) + \sum_{rr'} \min_{kk'} (g_{rr'}(k, k') + \varphi_{rr'}(k) + \varphi_{r'r}(k')) \rightarrow \max_{\Phi}$$
$$\text{s.t. } \varphi_r(k) + \sum_{r': rr' \in E} \varphi_{rr'}(k) = 0 \quad \forall r, k$$

eine konkave nicht überall differenzierbare Optimierungsaufgabe.

- Wie ist $SQ(A)$ (effizient) zu maximieren?
- Trivialität zu prüfen ist NP im Allgemeinen.
- Für welche A gibt es einen trivialen Äquivalent?

Wiederhole oft für alle r, k

- 1) Sammeln – gießen so viel wie möglich in $q_r(k)$:

$$\Delta_{rr'}(k) = \min_{k'} g_{rr'}(k, k')$$

$$q_r(k) = q_r(k) + \sum_{r': rr' \in E} \Delta_{rr'}(k, k')$$

$$g_{rr'}(k, k') = g_{rr'}(k, k') - \Delta_{rr'}(k, k')$$

- 2) Verteile gleichmäßig auf inzidente Kanten $g_{rr'}(k, k')$:

$$\Delta_r(k) = q_r(k)/4 \quad (\text{bei 4-Nachbarschaft})$$

$$g_{rr'}(k, k') = g_{rr'}(k, k') + \Delta_r(k)$$

$$q_r(k) = 0$$

Es ist nicht ganz klar, welche Aufgabe der Algorithmus eigentlich löst.
Im Allgemeinen wird SQ damit nicht global optimiert.

Praktisch funktioniert oft befriedigend.

Erweiterungen:

Message Passing Algorithmen – „gezielte“ (nicht gleichmäßige) Verteilung,
Subgradienten Verfahren ...

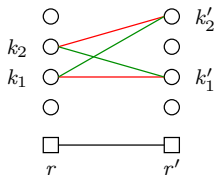
Polynomiell lösbare Spezialfälle

Im Allgemeinen sind diskrete Energieminimierung Probleme NP-vollständig.

Bekannte polynomiell lösbare Fälle:

- Der Graph der Aufgabe ist einfach, zum Beispiel eine Kette.
- Die Funktionen g haben bestimmte Eigenschaften.

Submodulare Aufgaben:



Sei die Menge der Label K vollständig geordnet, d.h. $K = \{1, 2, \dots, |K|\}$, sei $k_1 \leq k_2$ und $k'_1 \leq k'_2$ in dieser Ordnung. Die Funktion $g_{rr'}$ heißt submodular, wenn

$$g(k_1, k'_1) + g(k_2, k'_2) \leq g(k_1, k'_2) + g(k_2, k'_1)$$

für alle derartige viertupel k_1, k_2, k'_1, k'_2 .

Die Aufgabe heißt submodular, wenn alle Funktionen submodular sind.

Beispiele:

Entrauschung mit $(y_r - y_{r'})^2$ oder $|y_r - y_{r'}|$, manche binäre Segmentierungen usw.

Es gibt auch gemischte polynomiell lösbare Fälle.