Mustererkennung: Graphentheorie

D. Schlesinger – TUD/INF/KI/IS

Definitionen

Ein **Graph** ist ein Paar $G = (V, E)$ mit
der Menge der **Knoten** V und
der Menge der **Kanten**:

> **Gerichtete** Kanten – $E \subset V \times V$ – Teilmenge beliebiger (geordneter) Paare.
>
> **Ungerichtete Kanten** – $E \subset \{e = \{v, v'\} : v, v' \in V\}$ – Teilmenge der zweielementigen Teilmengen aus V,
> keine Schlingen (d.h. (v, v)) und keine doppelte Kanten (d.h. (v, v') und (v', v)).

Ein **Teilgraph** $G' = (V', E')$ eines Graphen $G = (V, E)$ ist ein Graph mit $V' \subset V$ und $E' \subset E$ (und natürlich $v \in e \in E' \Rightarrow v \in V'$).

Eine **Kette** ist ein Graph (mit endlich vielen Knoten), in dem jeder Knoten (bis auf zwei) genau zwei inzidenten Kanten hat.

Ein (einfacher) **Pfad** in einem Graphen G ist ein Teilgraph von G, der eine Kette ist.
Vereinfacht: eine Folge der Knoten $(v_1, v_2 \dots v_l)$ mit $\{v_i, v_{i+1}\} \in E$

Ein Graph heißt **zusammenhängend**, wenn für alle Paare v und v' ein Pfad von v nach v' existiert.

Ein **Zyklus** ist ein Graph, in dem alle Knoten genau zwei inzidenten Kanten haben.

Ein Graph ohne Zyklen (kein Teilgraph ist ein Zyklus) heißt **Wald**.
Ist der Graph zusammenhängend, so ist er ein **Baum**.

Kürzeste Pfade

Gewichtete Graphen: Gegeben ist eine Abbildung $w : E \to \mathbb{R}$, d.h. jeder Kante ist ein Gewicht $w(e) = w(\{r, r'\})$ zugeordnet.

Die **Kosten** eines Pfades ergeben sich als die Summe der Gewichte aller Kanten in dem Pfad.

Der **kürzeste** Pfad von v nach v' ist der Pfad minimaler Kosten
(heißt auch Abstand $d(v, v')$).

Drei Aufgaben:

a) Man such nach dem kürzesten Pfad von v nach v'

b) Man suche nach den kürzesten Pfaden von v nach allen anderen Knoten

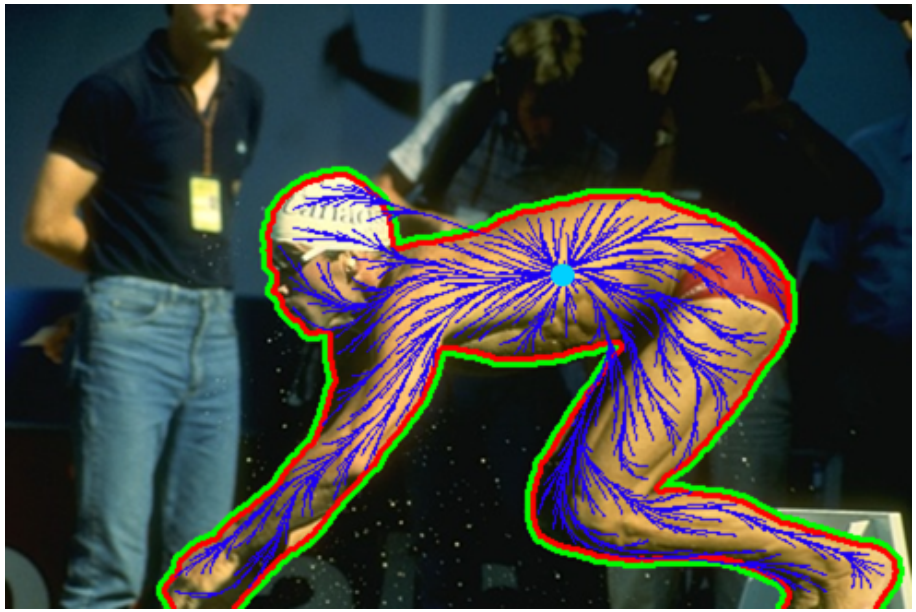
c) Man suche nach allen kürzesten Pfaden – Abstandsmatrix $d(v, v')$.

Paar Eigenschaften:

- Ist ein Pfad der kürzeste Pfad, so ist jeder zusammenhängender Teilgraph davon auch ein kürzester Pfad.
- Vereinigung aller kürzesten Pfade zu einem Knoten (Aufgabe b)) bilden einen Baum.

Kürzeste Pfade – Anwendungsbeispiel

„Geodesics“.



Die Knoten sind Pixel. Kanten verbinden nah liegende Pixel.

Die Kantengewichte berücksichtigen z.B. Farbdifferenz und Euklidische Länge

Die kürzesten Pfade von allen Pixeln zu einem (vom Nutzer gegebenem) werden berechnet.

Die Pixel werden als Vordergrund markiert, für die die Kosten des entsprechenden kürzesten Pfades kleiner als ein Schwellwert ist.

Algorithmus von Dijkstra

Gesuch wird nach den kürzesten Pfaden von einem ausgewählten Knoten v_0 zu allen anderen.

Hilfsmenge U:

Ist ein Knoten in V/U, so ist der kürzeste Pfad von v_0 zu diesem Knoten bereits berechnet.

Ist ein Knoten v noch in U, so bedeutet $d(v)$ die Kosten des kürzesten Pfades von v_0 nach v nur über Knoten in U/V.

1. Setze $U = V$, $d(v_0) = 0$, $d(v) = \infty$ für alle $v \in V/\{v_0\}$.
2. Falls $U = \emptyset$, dann STOP. Sonst weiter mit 3.
3. Finde ein $u \in U$, fur das $d(u)$ minimal ist.
4. Für alle $v \in U$ mit $\{u, v\} \in E$ setze $d(v) = \min\Big(d(v), d(u) + w(u, v)\Big)$.
5. Setze $U = U/\{u\}$. Gehe zu 2.

Zeitkomplexität: $\mathcal{O}(n^2 \log n)$

Alternative: Wiederhole oft für alle v:

$$d(v) = \min\Big[d(v), \min_{v' : vv' \in E} d(v') + w(v, v')\Big]$$

Schnitte

Betrachtet werden normalerweise nur zusammenhängende Graphen.

In der Menge der Knoten sind zwei ausgezeichnet – s (Quelle) und t (Empfänger).

Ein st-**Schnitt** ist eine Teilmenge der Kanten $C \subset E$ so, dass im Graphen $(V, E/C)$:

a) keinen Pfad von s nach t existiert

b) die Menge C ist nicht reduzierbar, d.h. ein Pfad von s nach t würde existieren, falls eine beliebige Kante von C entfernt wird.

In einem gewichteten Graphen sind die Kosten eines Schnittes die Summe aller Kantengewichte des Schnitts.

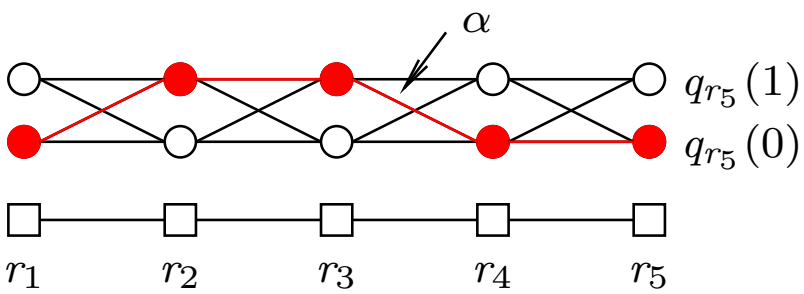
Gesucht wird nach dem Schnitt minimaler Kosten.

Sind alle Kantengewichte nicht negativ, ist die Aufgabe polynomiell lösbar (die Forderung b) ist redundant). Im allgemeinen Fall ist sie NP-vollständig.

Alternativ: Gesucht wird nach der Partitionierung der Menge der Knoten in zwei Teilmengen S und T (d.h. $S \cap T = \emptyset$, $S \cup T = V$) mit $s \in S$ und $t \in T$ so, dass

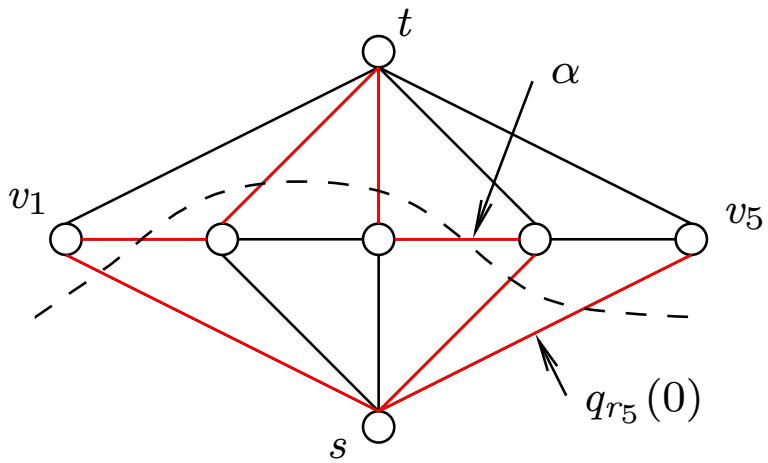
$$\sum_{vv' : v \in S, s' \in T} w(v, v') \to \min_{S,T}$$

Diskrete Energieminimierung als minimaler Schnitt



Potts Modell mit zwei Label:

$$E(y) = \sum_{r \in R} q_r(y_r) + \alpha \sum_{rr' \in E_R} \mathbb{I}(y_r \neq y_{r'})$$



Der Graph für Schnitt Aufgabe:
$G = (V, E_S)$ mit
$V = R \cup \{s\} \cup \{t\}$,
$E_S = E_R \cup \{s, v\} \cup \{v, t\}$
$w(s, v) = q_r(0),\ w(v, t) = q_r(1),$
$w(v, v') = \alpha$
Jeder Schnitt entspricht einem Labelling.
Die Kosten sind die Energien.

Beliebige diskrete Aufgaben der Energieminimierung (auch mit mehreren Labeln) lassen sich als Aufgaben des minimalen Schnittes formulieren.

Ist die ursprüngliche Aufgabe submodular, so sind alle Kantengewichte nicht negativ
$\rightarrow$ polynomiell lösbar.

Ein **Fluss** ist eine Abbildung $x : V \times V \to \mathbb{R}$, d.h. jeder gerichteten Kante (v, v') ist eine Zahl $x(v, v')$ zugeordnet (die Menge der Kanten wird verdoppelt). Dabei gilt:

$$\sum_{v' : vv' \in E} x(v', v) = \sum_{v' : vv' \in E} x(v, v')$$

für alle v, d.h. „wieviel in einen Knoten einfließt, genauso viel muss ausfließen“.

Weiterhin hat jede gerichtete Kante eine **Kapazität** $c(v, v')$ – vieviel darf durch die Kante maximal durchfließen.

Zwei Knoten s und t sind ausgezeichnet. Die Aufgabe besteht in der Suche nach maximalen Fluss von t nach s:

$$x(t, s) \to \max_x$$

s.t.

$$\sum_{v' : vv' \in E} x(v', v) = \sum_{v' : vv' \in E} x(v, v') \quad \forall v$$

$$x(v, v') \leq c(v, v') \quad \forall v, v'$$

Schnitte und Flüsse

Sind die Kapazitäten $c(v, v')$ gleich den (nicht negativen) Kantengewichten $w(v, v')$, so sind die entsprechenden Schnitt und Fluss Aufgaben zu einander **dual**.

- Der Werte des minimalen Schnittes und des maximalen Flusses sind gleich.
- Aus der Lösung der Fluss-Aufgabe (der maximale Fluss x d.h. alle entsprechenden $x(v, v')$) kann man den minimalen Schnitt C berechnen.

Algorithmus von Ford und Fulkerson $\rightarrow$ Seminar.

Empfehlenswert:
`http://www2.inf.fh-rhein-sieg.de/~pbecke2m/graphentheorie/`