

# Strukturelle Modelle in der Bildverarbeitung

## Markovsche Ketten III

D. Schlesinger – TUD/INF/KI/IS

- Wahrscheinlichste Zustandsfolge
- Bayessche Entscheidungstheorie
- Entscheidungsstrategien für HMM

Allgemein: Gegeben sei ein HMM

$$p(x, y) = p(y_1) \prod_{i=2}^n p(y_i | y_{i-1}) \prod_{i=1}^n p(x_i | y_i)$$

Man beobachtet  $x$ , man sage „etwas vernünftige“ über  $y$ .

Eine mögliche Wahl: Zustandsfolge, die die maximale a-posteriori Wahrscheinlichkeit hat:

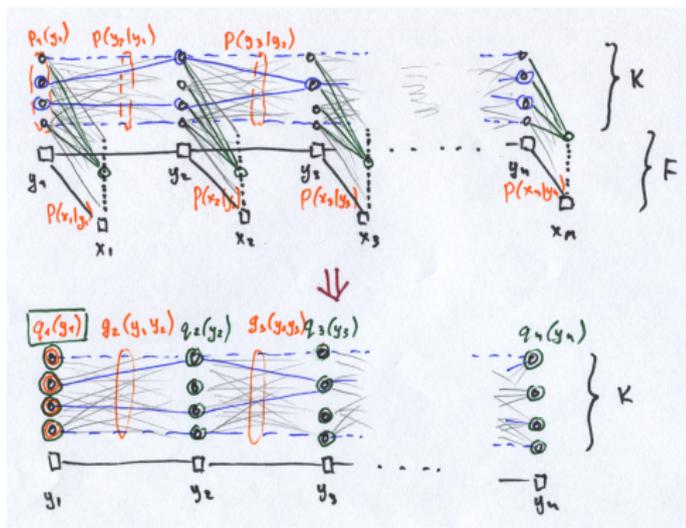
$$\begin{aligned} y^* &= \arg \max_y p(y|x) = \arg \max_y \frac{p(y)p(x|y)}{p(x)} = \arg \max_y p(y)p(x|y) = \\ &= \arg \max_y \left[ p(y_1) \prod_{i=2}^n p(y_i | y_{i-1}) \prod_{i=1}^n p(x_i | y_i) \right] = \\ &= \arg \min_y \left[ \sum_{i=1}^n q_i(y_i) + \sum_{i=2}^n g_i(y_i, y_{i-1}) \right] \end{aligned}$$

mit

$$q_1(k) = \ln p(k) + \ln p(x_1 | k) \quad q_i(k) = \ln p(x_i | k) \quad g_i(k, k') = \ln p(y_i | y_{i-1})$$

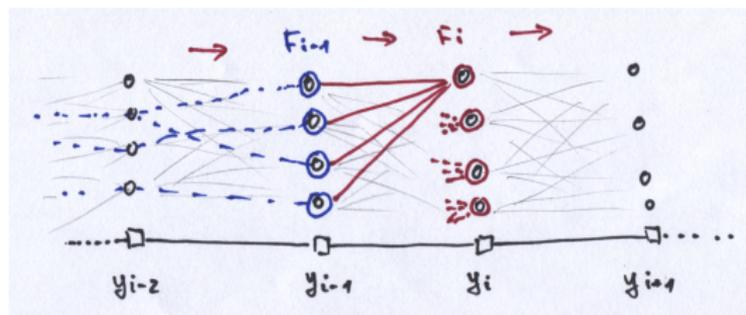
# Wahrscheinlichste Zustandsfolge

Der Algorithmus für die Berechnung von  $y^*$  – Dynamische Optimierung.  
Synonyme: Vitterbi, Dijkstra ...



Im Gegensatz zu Partition Funktion  
wird nicht über alle Folgen *summiert*, sondern *minimiert*  
die Bewertung einer Folge ist nicht ein *Produkt*, sondern eine *Summe*

Dasselbe wie bei der Berechnung der Partition Funktion, nur in einem anderen Semiring  
SumProd  $\Leftrightarrow$  MinSum



Die Idee – propagieren der Bellmanschen Funktionen  $F_i$  mit

$$F_i(k) = q_i(k) + \min_{k'} [F_{i-1}(k') + g_i(k', k)]$$

Sie repräsentieren die Kosten der besten Fortsetzungen auf den bereits bearbeiteten Teil des Problems.

$$\begin{aligned}
 & \min_y \left[ \sum_{i=1}^n q_i(y_i) + \sum_{i=2}^n g_i(y_{i-1}, y_i) \right] = \\
 & \min_{y_1} \min_{y_2} \dots \min_{y_n} \left[ \sum_{i=1}^n q_i(y_i) + \sum_{i=2}^n g_i(y_{i-1}, y_i) \right] = \\
 & \min_{y_2} \dots \min_{y_n} \left[ \sum_{i=2}^n q_i(y_i) + \sum_{i=3}^n g_i(y_{i-1}, y_i) + \min_{y_1} (q_1(y_1) + g_2(y_1, y_2)) \right] = \\
 & \min_{y_2} \dots \min_{y_n} \left[ \sum_{i=2}^n q_i(y_i) + \sum_{i=3}^n g_i(y_{i-1}, y_i) + F(y_2) \right] = \\
 & \min_{y_2} \dots \min_{y_n} \left[ \sum_{i=2}^n \tilde{q}_i(y_i) + \sum_{i=3}^n g_i(y_{i-1}, y_i) \right]
 \end{aligned}$$

mit  $\tilde{q}_2(k) = q_2(k) + F(k)$ , sonst unverändert, d.h.  $\tilde{q}_i(k) = q_i(k)$  für  $i = 3 \dots n$ .

```
// Forward pass
for  $i = 2$  bis  $n$ 
    for  $k = 1$  bis  $K$ 
         $best = \infty$ 
        for  $k' = 1$  bis  $K$ 
            if  $q_{i-1}(k') + g_i(k', k) < best$ 
                 $best = q_{i-1}(k') + g_i(k', k)$ ,  $pointer_i(k) = k'$ 
         $q_i(k) = q_i(k) + best$ 

// Backward pass
 $best = \infty$ 
for  $k = 1$  bis  $K$ 
    if  $q_n(k) < best$ 
         $best = q_n(k)$ ,  $x_n = k$ 
for  $i = n - 1$  bis  $1$ 
     $x_i = pointer_{i+1}(x_{i+1})$ 
```

$pointer_i(k)$  ist der beste Vorgänger für den Zustand  $k$  im  $i$ -ten Knoten.

# Dynamische Optimierung (Parallelisierung)

Zeitkomplexität:  $O(nK^2)$

$K$  Prozessoren:

die Schleife über  $k$  kann parallelisiert werden  $\rightarrow O(nK)$

Weitere Möglichkeit – mittlere Knoten Eliminieren.

$$\begin{aligned} & \min_{y_1} \min_{y_2} \min_{y_3} \left[ q_1(y_1) + g_2(y_1, y_2) + q_2(y_2) + g_3(y_2, y_3) + q_3(y_3) \right] = \\ & \min_{y_1} \min_{y_3} \left[ q_1(y_1) + q_3(y_3) + \min_{y_2} \left( g_2(y_1, y_2) + q_2(y_2) + g_3(y_2, y_3) \right) \right] = \\ & \min_{y_1} \min_{y_3} \left[ q_1(y_1) + q_3(y_3) + \tilde{g}(y_1, y_3) \right] \end{aligned}$$

$n/2$  Prozessoren:

die Eliminierungen können (fast) parallel ausgeführt werden  $\rightarrow O(\log n \cdot K^3)$

$n/2 \cdot K^2$  Prozessoren  $\rightarrow O(\log n \cdot K)$

Achtung!!! – etwas andere Bezeichnungen.

Gegeben sei zwei statistische Größen.

Typischerweise ist eine davon diskret (d.h.  $k \in K$ ) und heißt **Klasse**.

Die andere ist allgemein (sehr oft kontinuierlich, d.h.  $x \in \mathbb{R}^n$ ) und heißt **Beobachtung**.

„Gegeben“ sei die **Verbundwahrscheinlichkeitsverteilung**  $p(x, k)$ .

Da die Menge  $K$  diskret ist, wird oft  $p(x, k)$  durch  $p(x, k) = p(k) \cdot p(x|k)$  spezifiziert.

Die Erkennungsaufgabe:

man beobachtet  $x$ , man sage etwas über  $k$

– „welche Klasse hat die Beobachtung  $x$  verursacht“.

Für Markovsche Ketten ist die Menge der Klassen  $K$  die Menge aller Zustandsfolgen  $y$

Menge der Entscheidungen  $D$ , Entscheidungsstrategie  $e : X \rightarrow D$

Kostenfunktion  $C : D \times K \rightarrow \mathbb{R}$

Das Bayessche Risiko:

$$R(e) = \sum_x \sum_k p(x, k) \cdot C(e(x), k) \rightarrow \min_e$$

Spezialfall (fast immer) – die Menge der Entscheidungen ist nicht eingeschränkt:

$$R(e(x)) = \sum_k p(x, k) \cdot C(e(x), k) \rightarrow \min_{e(x)}$$

Noch spezieller (sehr oft) –  $D = K$

die Menge der Entscheidungen ist die Menge der Klassen:

$$k^* = \arg \min_k \sum_{k'} p(x, k') \cdot C(k, k')$$

Die Kostenfunktion ist (die einfachste)

$$C(k, k') = \mathbb{I}(k \neq k')$$

Daraus folgt die Maximum A-posteriori Entscheidung

$$\begin{aligned} R(k) &= \sum_{k'} p(k'|x) \cdot \mathbb{I}(k \neq k') = \sum_{k' \neq k} p(k'|x) \cdot 1 + p(k|x) \cdot 0 = \\ &= \sum_{k'} p(k'|x) - p(k|x) = 1 - p(k|x) \rightarrow \min_k \\ p(k|x) &\rightarrow \max_k \end{aligned}$$

# Additive Kostenfunktionen

Die Klasse ist ein Vektor  $\bar{k} = (k_1, k_2, \dots, k_n) \in K^n$ , die Entscheidungsmenge sei  $D = K^n$   
Die a-posteriori Wahrscheinlichkeit  $p(k_1, k_2, \dots, k_n | x)$  sei „bekannt“.

Variante 1: MAP, d.h.  $C(\bar{k}, \bar{k}') = \mathbb{I}(\bar{k} \neq \bar{k}')$

$$\bar{k}^* = (k_1, k_2, \dots, k_n)^* = \arg \max_{\bar{k}} p(k_1, k_2, \dots, k_n | x)$$

Variante 2: Kostenfunktionen gibt es für jeden Element  $k_i$ :

$$C(\bar{k}, \bar{k}') = \sum_i c_i(k_i, k'_i)$$

$$\begin{aligned} R(\bar{k}) &= \sum_{\bar{k}'} \left[ p(\bar{k}' | x) \cdot \sum_i c_i(k_i, k'_i) \right] = \sum_i \sum_{\bar{k}'} c_i(k_i, k'_i) \cdot p(\bar{k}' | x) = \\ &= \sum_i \sum_k \sum_{\bar{k}': k'_i = k} c_i(k_i, k) \cdot p(\bar{k}' | x) = \\ &= \sum_i \sum_k c_i(k_i, k) \sum_{\bar{k}': k'_i = k} p(\bar{k}' | x) = \sum_i \sum_k c_i(k_i, k) p(k'_i = k | x) \end{aligned}$$

$$R(\bar{k}) = \sum_i \sum_k c_i(k_i, k) p(k'_i = k | x) \rightarrow \min_{\bar{k}}$$
$$\sum_k c_i(k_i, k) p(k'_i = k | x) \rightarrow \min_{k_i}$$

1) Man berechne

$$p(k_i = k) = \sum_{\bar{k}: k_i = k} p(\bar{k}) \quad \forall i, k$$

2) Man treffe die Entscheidung für alle Elemente „unabhängig“

$$k_i^* = \arg \min_k \sum_{k'} p(k'_i = k) \cdot c_i(k, k')$$

Beispiele:

$c(k, k') = \mathbb{I}(k \neq k')$  – Max-Marginal Entscheidung  $k_i^* = \arg \max_k p(k'_i = k)$

$c(k, k') = (k - k')^2$  (mean square error) – Mittelwert  $k_i^* = \sum_k k \cdot p(k'_i = k)$

usw.