Viterbiho algoritmus

Viterbiho algoritmus je Maximum-Likelihood dekódování konvolučního kódu. Práce algoritmu je nejlépe pochopitelná pomocí mřížoví kódovače. Pro každý vrchol mřížoví s určí algoritmus poslední hranu nejpravděpodobnější cesty z počátečního vrcholu (nulového stavu v nulovém čase) do s. Výpočet probíhá rovnoměrně po časových vrstvách mřížoví. Do vrcholu nacházejícího se ve vrstvě t mřížoví vede 2^b hran z vrcholů z vrstvy $t-1$. (V příkladech budeme pro jednoduchost předpokládat $b = 1$ a $c = 2$.) Mezi nimi se vybere ta, která odpovídá pravděpodobnější cestě. Z toho je vidět, že náročnost algoritmu je dána počtem stavů kódovače.

Protože předpokládáme, že konec přenosu je opět ve stavu nula, získáme nakonec nejpravděpodobnější cestu z nulového stavu v nulovém čase do nulového stavu v čase t, kde t je délka přenosu (počet přechodů kódovače). Tuto cestu můžeme zrekonstruovat zpětným projitím mřížoví po vybraných hranách. Platí navíc, že cesty do všech stavů mají s velkou pravděpodobností společný začátek, takže dekódování může obvykle začít dříve než je dosažen konec vysílání.

Hledání nejpravděpodobnější cesty znamená, že chceme určit

$$
\begin{aligned}
\underset{\mathbf{v}}{\operatorname{argmax}} \Pr\left[\vec{r}_0 \vec{r}_1 \cdots \vec{r}_t | \vec{v}_0 \vec{v}_1 \cdots \vec{v}_t\right] &= \underset{\mathbf{v}}{\operatorname{argmax}} \prod_{i=0}^{t} \Pr\left[\vec{r}_i | \vec{v}_i\right] \\
&= \underset{\mathbf{v}}{\operatorname{argmax}} \prod_{i=0}^{t} \prod_{j=1}^{c} \Pr\left[r_i^{(j)} \mid v_i^{(j)}\right] \\
&= \underset{\mathbf{v}}{\operatorname{argmax}} \sum_{i=0}^{t} \sum_{j=1}^{c} \log \Pr\left[r_i^{(j)} \mid v_i^{(j)}\right] \\
&= \underset{\mathbf{v}}{\operatorname{argmax}} \sum_{i=0}^{t} \sum_{j=1}^{c} \mu(r_i^{(j)}, v_i^{(j)}),
\end{aligned}
$$

kde

$$
\mu(r_i^{(j)}, v_i^{(j)}) = a \cdot \left(\log \Pr\left[r_i^{(j)} \mid v_i^{(j)}\right] - f\left(r_i^{(j)}\right)\right)
$$

pro nějaké $a > 0$ a libovolné hodnoty $f(r) \in \mathbb{R}$. Zobrazení μ se nazývá *Viterbiho metrika* (nejedná se ovšem o metriku v topologickém smyslu). Hodnoty $f(r)$ i konstantu a volíme tak, aby se s metrikou dobře pracovalo, tedy nejlépe tak, aby její hodnoty byla malá přirozená čísla.

Je ještě třeba rozhodnout, jakých hodnot nabývají veličiny $r_i^{(j)}$. V případě tzv. *tvrdého dekódování* se jedná o prvky tělesa (tedy typicky 0 nebo 1). V případě *měkkého dekódování* se snažíme maximalizovat informaci získanou z kanálu, a rozlišujeme proto více možných hodnot a příslušné pravděpodobnosti. Pokud totiž vezmeme v úvahu, jak moc je přijatý (spojitý) signál podobný vyslanému (spojitému) signálu reprezentujícímu hodnotu 0 nebo 1, můžeme přisoudit větší důležitost těm hodnotám, které jsou jistější. Uvažme např., že tvrdé dekódování nás nutí přiřadit nulu nebo jeničku i signálu, který byl šumem natolik porušen, že jsou oba vyslané bity stejně pravděpodobné, a postavit ho tak na roveň jinému signálu, který je téměř neporušený. Měkké dekódování nám umožňuje tyto okolnosti zohlednit.

Ilustrujme to na příkladu binárního symetrického kanálu. V případě tvrdého dekódování stačí k charakteristice kanálu určit pravděpodobnost chyby ε. Tím vlastně charakterizujeme čtyři potřebné pravděpodobnosti:

$$\Pr[0 \mid 1] = \Pr[1 \mid 0] = \varepsilon, \qquad \Pr[0 \mid 0] = \Pr[1 \mid 1] = 1 - \varepsilon.$$

Položíme li nyní $f(0) = f(1) = \log \varepsilon$ a

$$a = \frac{1}{\log(1-\varepsilon) - \log \varepsilon},$$

dostáváme

$$\mu(d_1, d_2) = \begin{cases} 0 & \text{pokud } d_1 \neq d_2\,, \\ 1 & \text{pokud } d_1 = d_2\,. \end{cases}$$

Viterbiho algoritmus se tedy snaží maximalizovat počet shod. Jinak řečeno, jedná se o dekódování na nejbližší kódové slovo. Ověřili jsme celkem zřejmý fakt, že takové dekódování je ML.

V případě měkkého dekódování rozlišíme různé úrovně přijatého spojitého signálu, tento proces se nazývá *kvantizace*. Uvažujme například osmihodnotovou kvantizaci, jejíž hodnoty označme 0_1, 0_2, 0_3, 0_4, 1_1, 1_2, 1_3, 1_4, přičemž d_4 je signál nejbližší vyslanému $d \in \{0, 1\}$, a podobnost klesá s klesajícím indexem, takže d_1 je nejméně podobné vyslanému d (ale stále ještě podobnější d než $1 - d$). Přechodových pravděpodobností kanálu je nyní šestnáct, za předpokladu symetrie kanálu stačí uvést osm z nich. Např.:

$\Pr[r \mid v]$	d_4	d_3	d_2	d_1	$(1-d)_1$	$(1-d)_2$	$(1-d)_3$	$(1-d)_4$
b	0.434	0.197	0.167	0.111	0.058	0.023	0.008	0.002

Tedy např.

$$\Pr[0_3 \mid 0] = \Pr[1_3 \mid 1] = 0.197$$

nebo

$$\Pr[0_2 \mid 1] = \Pr[1_2 \mid 0] = 0.023\,.$$

Položíme-li nyní ve Viterbiho metrice

$$f(d_i) = \log \Pr[d_i \mid 1 - d]$$

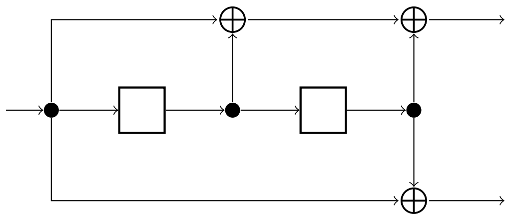
dostáváme po zaokrouhlení:

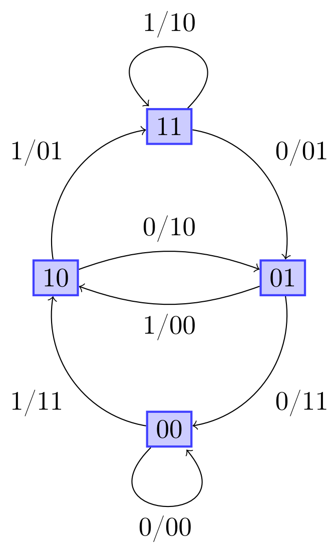
$\Pr[r \mid v]$	d_4	d_3	d_2	d_1	$(1-d)_1$	$(1-d)_2$	$(1-d)_3$	$(1-d)_4$
d	8	5	3	1	0	0	0	0

Rozdíl mezi tvrdým a měkkým dekódováním je možné ověřit na následujícím příkladě.

1. Příklad

$$G = (1 + D + D^2 \quad 1 + D^2)$$





$r = \quad 1_1 0_4 \qquad 0_1 1_2 \qquad 1_1 0_1 \qquad 0_1 0_1 \qquad 0_1 1_3 \qquad 0_4 0_3$