Aplikace fuzzy a pravděpodobnostních automatů

Martin Jašek

12. září 2016 — ??

Obsah

1 Definice a značení

Tato kapitola zatím poslouží jako „skladiště“ pro definice a zavedení značení pro ostatní kapitoly.

Abecedy, řetězce, jazyky

Abecedy budou značeny standardně, tedy velkými řeckými písmeny (typicky Σ). Řetězce pak malými písmeny ($\omega, \alpha, \dots$). Jazyky velkými kaligrafickými písmeny. Jazyk přijímaný automatem A bude značen $\mathcal{L}(A)$.

Fuzzy teorie

Fuzzy množiny a relace budou po vzoru [4] nejčastěji malými řeckými písmeny. Množinu všech fuzzy podmnožin množiny S budeme značit $\mathcal{F}(S)$.

Deterministický bivalentní automat

(zde bude doplněno: Značení převzato z FJAA, dohledat zdroj)

Definice 1.1. *Konečný deterministický (bivalentní) automat je pětice $A = (Q, \Sigma, \delta, q_0, F)$, kde Q je konečná množina stavů, Σ je vstupní abeceda, $\delta : Q \times \Sigma \to Q$ je přechodová funkce, $q_0 \in Q$ je počátační stav a $F \subseteq Q$ je množina koncových stavů.*

Nedeterministický bivalentní automat

(zde bude doplněno: Značení převzato z FJAA, dohledat zdroj)

Definice 1.2. *Konečný nedeterministický (bivalentní) automat je pětice $A = (Q, \Sigma, \delta, I, F)$, kde Q je konečná množina stavů, Σ je vstupní abeceda, $\delta : Q \times \Sigma \to 2^Q$ je přechodová funkce, $I \subseteq Q$ je množina počátačních stavů a $F \subseteq Q$ je množina koncových stavů.*

Základní definice nedeterministického fuzzy automatu

Značení je převzato z [4] a lehce upraveno.

Definice 1.3 (Nedeterministický fuzzy automat)**.** *Nedeterministický fuzzy automat A je pětice $(Q, \Sigma, \mu, \sigma, \eta)$, kde Q je konečná množina stavů, Σ je abeceda, μ je fuzzy přechodová funkce (fuzzy relace $Q \times \Sigma \times Q \to [0, 1]$) a σ a η jsou po řadě fuzzy množiny nad Q počátačních, resp. koncových stavů.*

Definice 1.4 (Fuzzy stav)**.** *Mějme nedeterministický fuzzy automat A. Pak jako fuzzy stav označujeme fuzzy podmnožinu jeho stavů, tj. $V \in \mathcal{F}(Q)$.*

Definice 1.5 (Aplikace fuzzy relace na fuzzy stav)**.** *Mějme nedeterministický fuzzy automat A a fuzzy symbol V. Pak aplikací binární fuzzy relace $R : Q \times Q \to [0, 1]$ na fuzzy stav V obdržíme fuzzy symbol $V \circ R$ splňující pro každé $p \in Q$: $(V \circ R)(p) = \max_{q \in Q}(V(p) \otimes R(p, q))$.*

Definice 1.6 (Přechodová funkce fuzzy stavů)**.** *Mějme nedeterministický fuzzy automat A. Pak přechodová funkce fuzzy stavů je fuzzy relace $\widehat{\mu} : \mathcal{F}(F) \times \Sigma \to \mathcal{F}(F)$ taková, že pro každý fuzzy stav $V \in \mathcal{F}(Q)$ a symbol $x \in \Sigma$ je $\widehat{\mu}(V, x) = V \circ \mu[x]$.*

Poznámka 1.1. *Označení $\mu[x]$ je fuzzy relace, pro kterou platí: $\mu[x](p,q) = \mu(p,x,q)$ pro každé $x \in \Sigma$.*

Definice 1.7 (Rozšířená přechodová funkce)**.** *Mějme nedeterministický fuzzy automat A. Pak rozšířená přechodová funkce (fuzzy stavů) je fuzzy relace $\mu^* : \mathcal{F}(F) \times \Sigma^* \to \mathcal{F}(F)$ daná následujícím předpisem:*

1. $\mu^*(V, \epsilon) = V$ *pro všechna* $V \in \mathcal{F}(Q)$
2. $\mu^*(V, x\alpha) = \widehat{\mu}(\mu^*(V, \alpha), x)$ *pro všechna* $V \in \mathcal{F}(Q), \alpha \in \Sigma^*, x \in \Sigma$

2 Rozpoznávání textových vzorů

Rozpoznávání vzorů obecně je jednou z nejvýznamějších aplikací informatiky. V běžném životě se často setkáváme se situacemi, kdy je třeba v datech najít výskyt učitého vzoru, popř. jeho další vlastnosti. Případně určit podobnost ke vzoru, nebo nejpodobnější vzor.

Typickým příkladem je např. detekce obličeje na fotografii, tedy rozpoznávání vzorů v obrazových datech. Vzory je však možné rozpoznávat v téměř jakýchkoliv datech, například textech, zvukových záznamech či výsedcích měření nebo pozorování.

Z pohledu teoretické informatiky je však základem vyhledávání vzorů v textových datech. Textová data, tedy řetězce, mají jednoduchou strukturu a lze s nimi snadno manipulovat. Na druhou stranu, jsou schopna reprezentovat nebo kódovat široké spektrum dat. Právě z tohoto důvodu je studium rozpoznávání textových vzorů klíčové pro zpracovávání jakýchkoliv dalších typů dat.

Poznámka 2.1. *Pokud nebude uvedeno jinak, pojem „rozpoznávání textových vzorů“ bude v této kapitole zkracován jen na „rozpoznávání vzorů“.*

2.1 Formální zavedení problému

Stejně tak, jak se mohou různit aplikace rozpoznávání vzorů, i samotný pojem „rozpoznávání vzorů“ bývá chápan různě. V nejzákladnější podobě se jedná o problém určení, zda-li pozorovaný řetězec odpovídá předem stanovenému vzoru. Vzorem bývá obvykle také řetězec, ale může jím být například regulérní výraz. Také - může nás zajímat buď exaktní shoda pozorovaného řetězce se vzorem, nebo jen nějaká forma podobnosti.

V rozšířeném smyslu může být problém chápán jako klasifikace. Tedy, určení třídy, do které by měl pozorovaný řetězec spadat, typicky na základě podobnosti s vybrannými reprezentanty jednotlivých tříd.

V této kapitole se však budeme zabývat pouze určováním podobnosti vzorového a pozorovaného řetězce. U každé instance problému budeme znát abecedu se kterou pracujeme a také vzor. Vzorem bude libovolný řetězec nad touto abecedou. Řešením tohoto problému pro nějaký, tzv. pozorovaný, vstupní řetězec bude úroveň podobnosti tohoto řetězce s vzorovým. Jako podobnost zde budeme uvažovat reálné číslo z intervalu $[0, 1]$, kde 0 znamená úplnou rozdílnost a 1 úplnou shodu.

Poznámka 2.2. *Vzorový řetězec budeme v této kapitole vždy značit ω, pozorovaný pak α.*

Nyní máme zadefinován problém samotný, nicméně je třeba zdůraznit, že v jeho definici se používá vágní pojem „podobnost řetězců“. Podobnost řetězců je totiž pojem, který souvisí s konkrétní instancí problému a nelze jej nějak přesně, ale současně dostatečně obecně popsat. Jedinné, co o podobnosti řetězců můžeme říct, je, že čím vyšší toto číslo je, tím by si měly být řetězce podobnější.

Například, budeme-li porovnávat vstup zadaný z klávesnice počítače oproti nějakému vzoru, je možné, že uživatel udělá překlep. V takovém případě bude vzorovému řetězci určitě více podobný řetězec obsahující dva překlepy (záměna symbolu za některý sousedící na klávesnici) než jiný, který se sice bude lišit jen v jednom symbolu, ale to takovém, který je na opačné straně klávesnice.

Obdobně, pokud budeme pracovat s abecedou malých a velkých písmen (majuskule a minuskule). Uvažujme vzorový řetězec `hello`. Řetězec `HELLO` se s ním neshoduje v ani jednom symbolu, ale přesto se jejich podobnost může blížit k jedné.

2.2 Motivace k použití fuzzy automatů

Klasická teorie automatů vznikla jako nástroj pro zpracování textových řetězců. Z tohoto důvodu je rozpoznávání textových vzorů jejím základním výsledkem. Automaty obecně jsou nástroje sloužící pro rozhodování, zda-li řetězec odpovídá vzoru automatem reprezentovaném. Použití pro rozpoznávání řetězcového vzoru tak bude jen speciálním případem jejich užití.

V předchozí podkapitole jsme si stanovili, že řešením našeho problému je číslo z intervalu $[0, 1]$. Z tohoto důvodu nebude možné využít klasické bivalentní automaty. Fuzzy automaty pracují se stupněm pravdivosti, který by mohl s hodnotou podobnosti řetězců korespondovat. Navíc, v praxi se často setkáme s texty, které jsou nepřesné a nedokonalé. Fuzzy přístup by nám tak mohl pomoci na tyto nepřesnosti adekvátně reagovat.

2.3 Automat rozpoznávající ω

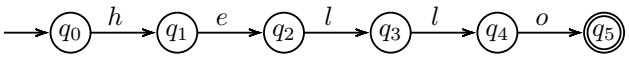
Klíčovým pro rozpoznávání vzorů (chceme-li využívat fuzzy automaty) je bivalentní automat rozpoznávající vzorový řetězec. Tedy automat takový, který přijímá jedinný řětězec ω a všechny ostatní zamítá. Nyní si takovýto automat zkonstruujeme.

Uvažujme, že máme k dispozici vzorový řetězec ω nad abecedou Σ. Označme $\mathcal{L}(\omega)$ jako jednoprvkový jazyk obsahující pouze řetězec ω. Vzhledem k tomu, že jazyk $\mathcal{L}(\omega)$ je konečný, je také regulérní a existuje tak konečný deterministický automat, který jej rozpoznává.

Automat bude v každém kroku konzumovat symboly ze vstupního řetězce a porovnávat je se symboly vzorového řetězce na odpovídajících pozicích. Pokud dojde ke shodě na všech pozicích, automat dojde do koncového stavu a sledovaný řetězec přijme. Pokud se symboly shodovat nebudou, automat nebude mít definován žádný odpovídající přechod, kterým by pokračoval ve výpočtu, a řetězec tak zamítne.

Takovýto automat označme jako *automat rozpoznávající ω*.

Definice 2.1 (Automat rozpoznávající ω (deterministický))**.** *Mějme řetězec ω délky n nad abecedou Σ. Automat rozpoznávající ω je pak konečný automat $A(\omega) = (Q, \Sigma, \delta, q_0, F)$ takový, že jeho množina stavů Q se sestává z právě n*



Obrázek 1: Automat rozpoznávající `hello`

stavů $q_0, \ldots, q_n$, q_0 *je počáteční stav,* $F = \{q_n\}$ *množina koncových stavů a* δ *je přechodová funkce definována pro všechna* $0 \leq k < n$ *následovně:*

$$\delta(q_k, a_k) = q_{k+1} \text{ kde } a_k \text{ je } k\text{-tý symbol řetězce } \omega$$

Tato definice automatu je vcelku intuitivní. K stejnému výsledku bychom došli, pokud bychom automat zkonstruovali konverzí gramatiky nebo regulérnmího výrazu. Příklad automatu rozpoznávající řetězec `hello` naleznete na obrázku 1.

Nyní tento automat převedeme na ekvivalentní fuzzy automat. Tedy na automat rozpoznávající řetězec ω ve stupni 1 a všechny ostatní ve stupni 0. Vzhledem k tomu, že budeme konstruovat fuzzy automat nedeterministický, bude vhodné si nejdříve přetransformovat deterministický automat z předchozí definice na nedeterministický.

Definice 2.2 (Automat rozpoznávající ω (nedeterministický))**.** *Mějme řetězec* ω *nad abecedou* Σ *z předchozí definice. Nedeterministický automat rozpoznávající* ω *je pak konečný automat* $A'(\omega) = (Q, \Sigma, \delta, I, F)$ *takový, že jeho množina stavů* Q *je stejná jako v předchozí definici, dále* $I = \{q_0\}$ *je množina počátečních a* $F = \{q_n\}$ *množina koncových stavů a* δ *je přechodová funkce definována pro všechna* $0 \leq k < n$ *následovně:*

$$\delta(q_k, a_k) = \begin{cases} \{q_{k+1}\} & \text{pokud je } a_k \text{ } k\text{-tý symbol řetězce } \omega \\ \emptyset & \text{jinak} \end{cases}$$

Definice 2.3 (Fuzzy automat rozpoznávající ω)**.** *Mějme řetězec* ω *nad abecedou* Σ *délky* n *z předchozí definice. Automat rozpoznávající* ω *je pak fuzzy automat* $A''(\omega) = (Q, \Sigma, \mu, \sigma, \epsilon)$ *takový, že jeho množina stavů* Q *je stejná jako v předchozí definici, a dále*

- $\sigma(q_0) = 1$ *a* $\sigma(q_i) = 0$ *pro všechna* $i > 0$
- $\epsilon(q_n) = 1$ *a* $\epsilon(q_i) = 0$ *pro všechna* $i < n$
- $\mu(q_k, a_k, q_{k+1}) = \begin{cases} 1 & \text{pokud je } a_k \text{ } k\text{-tý symbol řetězce } \omega \\ 0 & \text{jinak} \end{cases}$

Nyní máme k dispozici fuzzy automat, který ostře rozpoznává vzorový řetězec. V následujících podkapitolách následuje výčet několika technik, které tuto ostrost odstraňují a nahrazují podobností.

2.4 Fuzzy symboly

Nejzákladnějším způsobem, jak zanést neurčitost do rozpoznávání vzorů, je použití fuzzy symbolů. Idea pro použití fuzzy symbolů byla přejata z [2] a vychází z [1].

Fuzzy symbol je nástroj, který akceptuje nepřesnost na úrovni jednotlivých symbolů. Fuzzy symbol reprezentuje relaci podobnosti symbolů.

Mějme abecedu Σ a pro každé dva symboly z této abecedy číslo z intervalu $[0, 1]$ udávající jejich podobnost. Například, symbol reprezentující malé psací `i` bude mít vyšší podobnost s malým psacím `e` než s malým psacím `m`. Tuto podobnost můžeme realizovat jako fuzzy relaci nad $\Sigma \times \Sigma$. Tento způsob je použit v [1]. Vzhledem k tomu, že očekáváme použití automatů, bude však vhodnější držet se symbolů a abeced. Zavádí se proto speciální třída symbolů, tzv. fuzzy symboly, a s nimi související pojmy.

Definice 2.4 (Fuzzy symbol [3])**.** *Mějme abecedu Σ a nějaký symbol y z této abecedy. Fuzzy symbol $\widetilde{y}$ symbolu y je fuzzy množina nad abecedou Σ popisující podobnost symbolů. Tedy, pro každé $x, y \in \Sigma$ je $\widetilde{y}(x)$ rovna stupni podobnosti x a y. Mělo by platit, že $\widetilde{y}(y) = 1$.*

Máme-li definován fuzzy symbol pro všechny symboly y z abecedy Σ, můžeme tak nadefinovat abecedu fuzzy symbolů.

Definice 2.5 (Abeceda fuzzy symbolů)**.** *Mějme abecedu Σ a fuzzy symboly $\widetilde{y}$ pro všechna $y \in \Sigma$. Pak množinu všech těchto fuzzy symbolů nazvěme abeceda fuzzy symbolů a označme $\widetilde{\Sigma}$. Tedy $\widetilde{\Sigma} = \{\widetilde{y} \mid y \in \Sigma\}$.*

Nyní máme k dispozici abecedu pracující s fuzzy symboly. S touto abecedou můžeme pracovat jak jsme zvyklí, tedy používat operace definované nad abecedami. Máme tak také možnost sestrojit řetězec fuzzy symbolů nad touto abecedou. Pro náše účely bude výhodné, abychom dokázali převést řetězec nad abecedou Σ na odpovídající řetězec nad abecedou $\widetilde{\Sigma}$.

Definice 2.6 (Řetězec fuzzy symbolů)**.** *Mějme abecedu Σ a nějaký řetězec $\alpha = a_1 \dots a_n$ nad touto abecedou (tj. $\alpha \in \Sigma^*$). Pak definujme $\widetilde{\alpha} = \widetilde{a_1} \dots \widetilde{a_n}$ jako řetězec fuzzy symbolů řetězce α.*

Nyní máme nadefinováno vše potřebné a můžeme tedy zkonstruovat automat, který rozpoznává vzorový řetězec, a to s ohledem na podobnost symbolů. Idea je jednoduchá – vezmeme fuzzy automat rozpoznávající ω, jeho abecedu nahradíme abecedou fuzzy symbolů a patřičně předefinujeme některé další pojmy.

Definice 2.7 (Automat pracující s fuzzy symboly)**.** *Mějme abecedu Σ. Pak na základě podobností symbolů v Σ sestavíme abecedu $\widetilde{\Sigma}$ fuzzy symbolů. Pak automat pracující s fuzzy symboly je fuzzy automat definovaný v definici 1.3, ve které je Σ nahrazena $\widetilde{\Sigma}$.*

U takovéhoto automatu pochopitelně dojde ke změne způsobu výpočtu. Proces jeho výpočtu se tak změní ve fázi výpočtu přechodové funkce fuzzy stavů. Připomeňme, že ta je definována (definice 1.7) jako fuzzy relace $\widehat{\mu}$ přiřazující každému fuzzy stavu V a fuzzy symbolu x fuzzy stav dle předpisu

$$\widehat{\mu}(V, x) = V \circ \mu[x]$$

Zde je zjevně nutné nahradit $\mu[x]$ spojením přes všechny fuzzy symboly. Bude tedy vypadat následovně:

$$\widehat{\mu}(V, x) = V \circ \bigvee_{y \in \Sigma} (\mu[x] \wedge \widetilde{x}(y))$$

Nyní můžeme zkonstruovat automat rozpoznávající ω pracující s fuzzy symboly.

Definice 2.8 (Automat rozpoznávající ω pracující s fuzzy symboly)**.** *Mějme abecedu Σ a vzorový řetězec ω. Pak na základě podobností symbolů v Σ sestavíme abecedu $\widetilde{\Sigma}$ fuzzy symbolů. Pak automat pracující s fuzzy symboly je fuzzy automat rozpoznávající ω definovaný v definici 2.3, ve které je Σ nahrazena $\widetilde{\Sigma}$.*

Nyní máme zkonstruován požadovaný automat. Podíváme-li se na jeho rozpoznávací charakteristiku, je zjevné, jaké řetězce tento automat rozpoznává. Tento automat rozpoznává vzorový řetězec ve stupni 1 (za předpokladu, že byla dodržena vlastnost $\widetilde{y}(y) = 1$ pro všechna $y \in \alpha$). Při zpracování jakéhokoliv jiného řetězce automat nezkončí v koncovém stavu. Nicméně, díky pozměněnému výpočtu přechodové funkce fuzzy stavů je i v takové situaci schopen řetězec přijmout v nenulovém stupni. K tomu dojde právě v případě, kdy se vzorový a pozorovaný řetězec na odpovídajících pozicích liší, ale tyto symboly jsou si podobné. Připomeňme, že tato podobnost je dána hodnotou stupně fuzzy symbolu.

Automat tak dobře zvládá rozpoznat řetězce, u kterých došlo k záměně podobným symbolem. Na druhou stranu, tato technika se nehodí na situace, kdy by mohl být do řetězce vložen, nebo naopak odebrán symbol (nebo více symbolů).

Reference

[1] H. A. Girijamma Dr. V. Ramaswamy. Conversion of finite automata to fuzzy automata for string comparison. *International Journal of Computer Applications*, 2012.

[2] J. J. Astrain; J. R. Garitagoitia; J. R. Gonzalez De Mendivil; J. Villadangos; F. Farina. Approximate string matching using deformed fuzzy automata: A learning experience. *Springer Science & Business Media*, 2004.

[3] J R G; Echanobe J; Astrain J J; Farina F. Garitagoitia, J R; de Mendivil. Deformed fuzzy automata for correcting imperfect strings of fuzzy symbols. *IEEE Transactions on Fuzzy Systems*, 2003. K dispozici na IEEExplore.

[4] J.R. Garitagoitia J. Astrain, J.R. González de Mendívil. Fuzzy automata with ϵ-moves compute fuzzy measures between strings. *Fuzzy Sets and Systems 157*, 2005.