

# Klasifikace XML dokumentů

Martin Procházka a Jan Blažák

Laboratoř vyhledávání znalostí  
Fakulta informatiky  
Masarykova Univerzita v Brně, Česká republika  
{xproch11,xblatak}@fi.muni.cz

**Abstrakt.** V tomto článku prezentujeme novou metodu pro klasifikaci XML dokumentů, která využívá nejen vlastní data uložená v dokumentu, ale také jeho strukturu. Přitom však nevyžaduje dodatečné informace jako XML schéma nebo DTD. Je navržena nová metoda pro transformaci XML dat do podoby jediné tabulky, kterou lze poté zpracovat stávajícími systémy strojového učení. Uvedeme analýzu výsledků experimentů na dokumentech vytvořených z Internet Movie Database (IMDb).

**Klíčová slova:** XML, XML mining, schemaless XML classification

## 1 Úvod

XML (eXtensible Markup Language) [9] je otevřený standard pro definování vlastního strukturovaného jazyka (tzv. XML dialektu), představujícího formát pro ukládání informací. Tuto definici představuje XML schéma nebo DTD soubor, který se pak využívá pro ověření správnosti (validaci) zpracovávaného dokumentu. Data uložená v XML pak těží zejména z výhod otevřených standartů a snadné výměny mezi aplikacemi. XML dokumenty pak navíc můžeme reprezentovat pomocí stromové struktury, která umožňuje členit dokument na jednotlivé části a poskytuje další užitečné informace. Protože XML umožňuje modelovat složité závislosti, může být použito i pro ukládání dat s členitou strukturou.

Nabízí se dvě možné cesty jak pracovat s XML daty: transformovat je do podoby vhodné pro nějaký zavedený systém pro dolování znalostí (hovoříme pak o tzv. propozičních datech), nebo adaptovat stávající systémy tak, aby mohly zpracovávat přímo XML dokumenty. V tomto článku podáme přehled stávajících metod pro dolování v XML a uvedeme novou metodu pro transformaci XML dokumentů a jejich klasifikaci. Metodu experimentálně ověříme na datech z internetové filmové databáze (Internet Movie Database, IMDb<sup>1</sup>) a porovnáme ji s předešlými přístupy.

## 2 Dolování v XML

Dolování v XML bylo až donedávna neprozkoumanou oblastí a dalo by se říci, že tomu tak stále ještě je. XML často zpracováváme tak, že provedeme extrakci

<sup>1</sup> <http://www.imdb.com>

vlastních dat na základě našich potřeb a znalosti konkrétního dialektu. Získáme jedinou tabulku, kterou předložíme propozičnímu učicímu systému. Hovoříme pak o tzv. propozicionalizaci [5]. Převod dat však můžeme do jisté míry provést automaticky, např. pomocí systémů pro multirelační dolování znalostí [2]. Vzhledem ke stále rostoucímu počtu XML dokumentů však roste potřeba metod, které pracují přímo s XML dokumenty a využívají všech informací, které jsou v nich obsaženy – tedy i těch o struktuře.

Pro dolování ve struktuře XML navrhl Termier a kol. [7] algoritmus TreeFinder, který hledá pomocí upraveného algoritmu Apriori [1] nejčastěji se vyskytující nejspecifičtější podstromy. Ve výsledném podstromu jsou zachovány tranzitivní vztahy předek–potomek ale ne nutně rodič–dítě. Algoritmus nejprve z každého vstupního XML dokumentu vytvoří transakci, která obsahuje všechny možné kombinace (element–element). Elementy jsou přitom buď ve vztahu rodič–dítě a nebo předek–potomek. V této množině se pomocí AprioriTree naleznou nejčastější relace a z nich se znovu sestaví nejméně obecný strom.

Zaki a kol. [10] navrhl metodu XRules pro klasifikaci XML dokumentů. Vychází z algoritmu TreeFinder, na jehož základě je vytvořen algoritmus XMINER, pro hledání tzv. klasifikačních asociačních pravidel [6] (asociační pravidlo s identifikátorem třídy v závěru) v XML dokumentech. Z pravidel, která jsou splněna pro klasifikovanou instanci, se vybírá podmnožina se shodnou třídou a nejvyšším *kombinovaným efektem* (pokrytí, spolehlivost a korelace). Nevýhodou tohoto přístupu pak může být fakt, že celá klasifikace je prováděna pouze na základě struktury XML.

Proto Theobald a kol. [8] vyvinul metodu převodu XML na propoziční data, která využívá jak části strukturní informace tak dat uložených v dokumentu. Nové rysy jsou vytvářeny jako kombinace vlastních dat a základní strukturní informace: atribut–hodnota, element–atribut, element–termín, nebo využívají pouze strukturu: element–element (rodič–dítě), element–element–element (levé dítě–rodič–pravé dítě). Problém různého pojmenování elementů v dokumentech z různých zdrojů byl řešen mapováním ontologií. Metodu úspěšně použili na klasifikaci dat z IMDb. Na tuto metodu navazujeme při návrhu nové metody prezentované v tomto článku.

### 3 Klasifikace dat v XML

Klasifikace je jednou z nejčastěji používaných úloh dolování znalostí. Uplatňuje se totiž v mnoha doménách, např. při filtrování spamů (klasifikace dokumentů). Většinu stávajících systémů strojového učení však nelze použít přímo na data v XML. Mohli bychom sice použít některý systém pro relační dolování [2], ty ale vyžadují definici doménové znalosti a jejich použití je tak dost náročné. Nejjednodušší by bylo informaci o struktuře úplně pominout a klasifikovat data jako prostý text. Tím se však připravíme o informace, které mohou pomoci při analýze dat. Vhodnější je proto transformovat data na jedinou tabulku takovou metodou, která zachová alespoň část strukturní informace. Následující text věnujeme popisu metody pro klasifikaci XML dokumentů založenou na

tomto postupu. Dokumenty jsou tedy nejprve převedeny do jediné tabulky (viz 3.1), ze které jsou poté vybrány pouze významné rysy (viz 3.2). Navržený systém jsme implementovali jako skript v jazyce Python a nyní je dostupný na <http://www.fi.muni.cz/~xproch11/xml>.

### 3.1 Transformace dat

Při návrhu metody jsme vycházeli zejména z prací M. Theobalda a kol. [8] a A. Termiera a kol. [7]. Hlavní rozdíl proti Theobaldovu přístupu spočívá ve způsobu vytváření rysů nesoucích informaci o struktuře dokumentu. Zavedli jsme nové rysy, které jsou obecnější a jsou proto schopny lépe zpracovat různě strukturované dokumenty. Jako příklad můžeme uvést rys typu *předek–potomek*, který je schopen popsat dokumenty, ve kterých jsou stejné elementy zapsány jednou jako řetězec a podruhé ve formě stromu.

Vytvářené rysy můžeme rozdělit do tří skupin podle typu nesené informace: *strukturní*, *strukturně-datové* a *datové* (termíny).

#### Strukturní rysy

**element–element:** daný element se vyskytuje v dokumentu (např. `movie@movie` nebo `genre@genre`)

**element–element (rodič–dítě):** zachycuje dva uzly spojené hranou (např. `movie@genres`, `genres@genre` nebo `roles@role`)

**element–element (předek–potomek):** všechny tranzitivní relace od kořene směrem k listům (k rysům z předchozího příkladu by přibylo `movie@genre` nebo `movie@role`).

#### Strukturně-datové rysy

**element–atribut:** kombinace jména elementu a jména atributu, který je v něm obsažen (např. z `<role actor="Bruce Willis">` bude výsledkem `role@actor`)

**element–atribut–hodnota:** ke jménu elementu a atributu přibývá hodnota atributu (např. `role@actor@Bruce Willis`)

**atribut–hodnota:** spojení jména atributu a jeho hodnoty (`actor@Bruce Willis`)

**element–hodnota:** se jménem elementu je spojena hodnota atributu, který je v něm obsažen (např. `role@Bruce Willis`)

**element–termín:** spojení jména elementu s termínem, kde termíny jsou řetězce oddělené neviditelnými znaky z oblasti uzavřené elementem (z `<role>Man in coon skin hat</role>` dostaneme `role@Man`, `role@in`, `role@coon`, `role@skin`, `role@hat`).

#### Datové rysy

**termín:** jedná se o zobecnění rysu typu *element–termín*, kde termín není vázán na nadřazený element (dostaneme např. rysy `@Man`, `@in`, tedy vlastní data).

### 3.2 Výběr rysů

Počet vytvořených rysů je většinou příliš vysoký a jen malá část z nich je přitom vhodná k sestavení klasifikátoru. Ze sta dokumentů lze zkonstruovat i několik tisíc rysů, nicméně jejich velká část pokrývá velmi malé množství dokumentů.

Abychom urychlili fázi učení, používáme v navržené metodě pro výběr významných rysů filtrování [4]. Jako metriku jsme použili *informační zisk* (*information gain*, IG), která je založena na principu minimalizace entropie. Míra se ukázala jako velmi vhodná pro různé typy dat [3]. Hodnota této metriky tedy byla použita pro uspořádání všech vytvořených rysů, ze kterých se poté vybíralo prvních  $N$ , kde  $N$  zadává uživatel.

## 4 Experimenty

Metodu jsme testovali na datech z IMDb, ze kterých byly vytvořeny nové XML dokumenty. Data jsme získali od autorů práce [8]. Tato databáze obsahuje název, rok výroby, žánry, seznam herců a řadu dalších informací o filmech. Data přitom nemusejí být úplná. U mnoha filmů z počátku dvacátého století, především pak u němých snímků, není k dispozici seznam rolí a herců, kteří je ztvárňují. Naproti tomu u nových titulů máme k dispozici nejen úplný popis rolí, ale také realizační štáb či krátký popis zápletky a odkazy na příbuzná díla.

Úloha spočívala v klasifikaci filmů do dvou tříd. Třídy byly definovány na základě žánrů a tato informace byla samozřejmě z dat odstraněna. Zavedli jsme třídy komedie, drama, western a akční film. Z každé třídy bylo náhodně vybráno 50 filmů, které jsou reprezentovány samostatnými XML dokumenty.

Testovali jsme čtyři různé metody transformace dat rozdělené podle použitého typu rysů: CDATA (datové rysy), STRUCT (strukturní rysy), STRUCT + CDATA (strukturní a datové rysy) a ALL (všechny typy rysů). Vybírali jsme vždy prvních 1 až 5, 10, 20, 40, 60, 80, 100, 200, 300, 400, 500 a 1 000 nejlepších rysů. Jako hodnoty atributů byly ve výsledné tabulce použity hodnoty IG konkrétních rysů. Pro klasifikaci jsme použili algoritmy Naïve Bayes, IBk, SVM a J4.8 ze systému WEKA<sup>2</sup> s implicitním nastavením. Učení a testování bylo prováděno desetinasobnou křížovou validací s 90 trénovacími a 10 testovacími příklady.

## 5 Výsledky

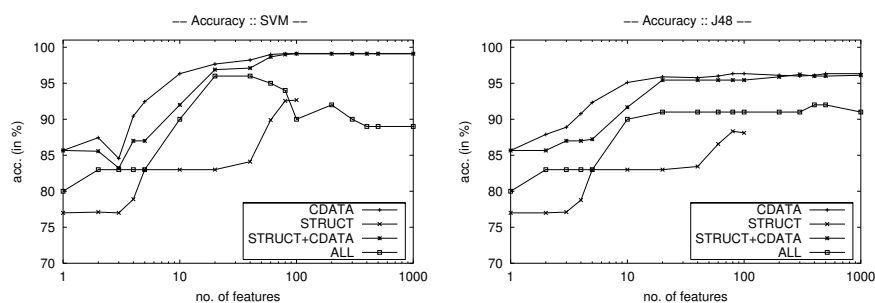
**Westerny proti akčním filmům.** Klasifikace akčních filmů proti westernům byla řešena také v práci M. Theobalda a kol. Použili jsme proto pro naše experimenty stejné nastavení (to znamená stejný počet učicích a testovacích příkladů), abychom mohli porovnat dosažené výsledky.

Při transformaci dat bylo vytvořeno 4 795 (CDATA), 105 (STRUCT), 4 900 (STRUCT+CDATA) a 15 170 (ALL) rysů. Pro názornost uvádíme zástupce jednotlivých typů rysů. Jedná se o vytvořené rysy, které získaly vysoké hodnoty informačního zisku.

<sup>2</sup> <http://www.cs.waikato.ac.nz/ml/weka/>

movie@soundmix	(rodič-potomek)
colourinfo@and	(element-termín -- black and white)
soundmix@soundmix	(element-element)
plot@author@maurice	(element-atribut-hodnota)
plot@maurice	(element-hodnota)
@silent	(termín)

Závislost přesnosti klasifikace učičů SVM a J4.8 na typu a počtu použitých rysů zobrazuje levý graf <sup>3</sup> na obr. 2. Z grafů je patrné, že klasifikace na základě vlastních dat (CDATA) je lepší, nebo stejně dobrá, jako při použití nových rysů. Graf na obr. 2 ukazuje vývoj hodnoty  $F_1$  míry. S referenční metodou [8] jsme



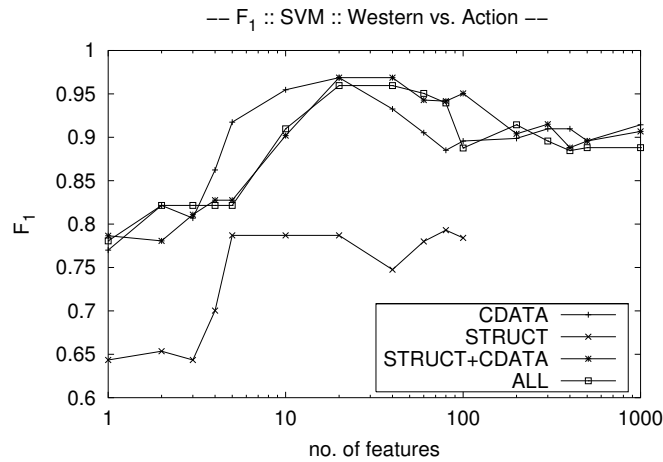
**Obr. 1.** Závislost přesnosti klasifikace na typu a počtu použitých rysů.

srovnávali právě přes touto metrikou vypočtené hodnoty pro algoritmus SVM. Autoři práce ukázali, že jejich metoda dosahuje lepších výsledků než klasifikace pomocí pouhých dat. Maximální hodnota  $F_1$  míry se přitom pohybuje kolem 0.85 pro 100 a více rysů. V našich experimentech jsme dosáhli celkově vyšší přesnosti a to i při menším počtu rysů. Například již pro 10 rysů jsme naměřili 0.91 a maxima (0.96) jsme dosáhli při použití pouhých 20 rysů. Závislost přesnosti klasifikace zbylých dvou učičů (IBk a NB) na počtu rysů je podobná jako u SVM, nedosahují však tak dobrých výsledků.

Musíme tu však poznamenat, že nejlepší výsledky byly dosaženy pro CDATA (viz obr. 1), což je způsobeno výskytem silných datových rysů, které velmi dobře rozdělují data. Vyšší přesnost oproti referenční metodě pak byla s největší pravděpodobností dosažena díky použití míry IG namísto MI-score pro výběr rysů. Tato zjištění však ještě neznamenají, že se strukturní rysy při klasifikaci nemohou uplatnit.

**Komedie proti dramatu.** Abychom skutečně ověřili kvalitu námi navržené metody, vybrali jsme z databáze IMDb data, která není možné přesně klasifikovat

<sup>3</sup> Všechny grafy jsou zobrazeny pro lepší přehlednost s použitím logaritmické škály na ose  $x$ .



**Obr. 2.** Závislost přesnosti klasifikace a hodnoty míry  $F_1$  na typu a počtu použitých rysů. Jako pozitivní třída byly zvoleny westerny stejně jako v [8].

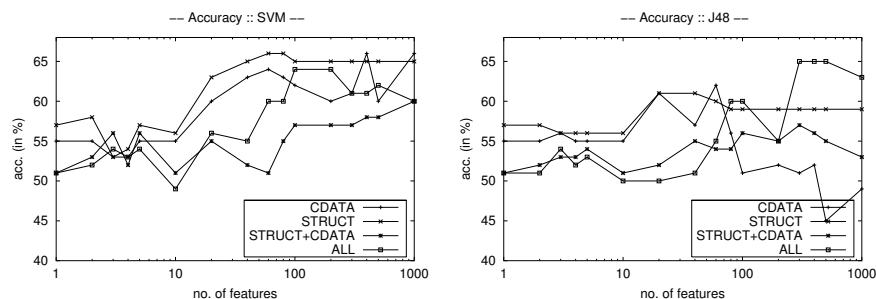
pomocí jednoduchých rysů. Vybrali jsme proto filmy označené jako komedie a drama, na kterých klasifikátor SVM dosáhl nejvíce 66% přesnosti (při výběru 400 a 1000 rysů). Při výběrů příkladů jsme pak volili takové filmy, u kterých bylo uvedeno co nejvíce možných informací. Opět jsme pak testovali všechny čtyři metody tvorby rysů.

Přes špatné výsledky v přesnosti na této úloze, je z grafů na obrázku 3 vidět, že strukturní rysy mají pozitivní vliv na přesnost klasifikace složitějších dat. Byly zkonstruovány rysy představující zajímavé znalosti. Komedie měly často víceslovné názvy a tím pádem byly mezi významnými rysy `title@and`, `title@a`, `atp`.

Strukturní rysy vedly ke zvýšení přesnosti již při malém počtu atributů. Hodnoty 66 % bylo systémem SVM dosaženo již při klasifikaci dat obsahujících pouhých 60 rysů. Při klasifikaci dat systémem J4.8 se při tomto počtu rysů zase nejlépe osvědčila metoda CDATA, přestože nejlepší dosažený výsledek tohoto učiče je s metodou ALL při 300 až 500 rysech. Systémy IBk a NB dosáhly shodně nejvyšší přesnosti při metodě ALL a 200 rysech, přičemž jsou celkově až o 5 % horší než SVM nebo J4.8.

## 5.1 Diskuze

Důvodem proč strukturní a strukturně-datové rysy nezvyšují vždy přesnost, může být jejich velká specifičnost. Tím vysvětlujeme celkově menší přesnost metody ALL, kdy byly zkonstruovány rysy, které vedly k přeučení. Například rys `casting@position@3` (`element-atribut-hodnota`) může mít vysoké IG na trénovacích datech, ale není obecný a povede ke snížení přesnosti na testovacích datech.



**Obr. 3.** Závislost přesnosti klasifikace dramatu vůči komediím na typu a počtu použitých rysů.

Pokud bychom tedy nahlíželi na XML dokument jako na množinu stejně významných termínů, zjistíme, že obecně je přesnost v klasifikaci při malém počtu rysů vyšší u strukturu-využívajících metod, ale při vyšším počtu rysů je rozdíl zanedbatelný. Samotné termíny ale mohou být u složitějších a objemnějších dat příliš obecné a struktura může plnit funkci váhování, kdy element ve kterém se termín nachází, představuje kontext (např. `colourinfo@black` vůči `plot@black`). Jak je vidět z grafů na obrázku 3, mají strukturní rysy pozitivní vliv na přesnost při klasifikaci složitějších dat. Při klasifikaci westernů a akčních filmů vítězí vlastní data nad strukturou dokumentů. Je to způsobeno tím, že jsou informace u obou typů filmů velmi odlišné. Pro klasifikaci se tak nejlépe uplatnily rysy identifikující němý nebo černobílý film. Naproti tomu se při klasifikaci komedií vůči dramatům prosadí rysy založené na struktuře, protože se v samotných datech nevyskytují termíny, které by jednoznačně určovaly výslednou třídu.

Zjistili jsme, že používat více než tisíc rysů nemá význam, protože v nich už nenajdeme takové, co by obsahovaly informaci, která pomůže zvýšit přesnost, ale naopak zkomplikují proces učení.

## 6 Závěr

Navrhli a implementovali jsme metodu pro klasifikaci XML dokumentů bez použití schématu. Provedli jsme experimenty na XML dokumentech vytvořených z IMDb a uvedli srovnání s referenční metodou.

Dva prezentované experimenty ukazují protikladné typy problémů, se kterými se můžeme setkat. V prvním experimentu, který jsme převzali z prezentace referenční metody, nezvýšily strukturní rysy přesnost klasifikace. Zjistili jsme, že informačně silné rysy, které popisovaly parametry média (barva, zvuk, atp.), a které se dají získat reprezentací samotného dokumentu jednoduchým vektorem, vedou k lepší přesnosti. Druhý experiment (komedie proti dramatu) ukazuje, že užití struktury má význam až u složitějších dat, kdy samotné termíny jsou při-

liš obecné a jejich váhování (např. spojení jména elementu a termínu) vede ke zvýšení přesnosti klasifikace.

## 7 Poděkování

Náš největší dík patří autorům článku [8] (M. Theobald a kol.), kteří nám laskavě k experimentování poskytli XML verzi databáze IMDb. Rádi bychom také poděkovali L. Popelínskému a anonymním recenzentům za cenné rady a připomínky. Tato práce je částečně podporována z grantů MŠMT 143300003 a MSM 0021622418.

## Reference

1. Agrawal R. and Srikant R. Fast algorithms for mining association rules in large databases. In Bocca J. B., Jarke M., and Zaniolo C., editors, *VLDB'94, Proc. of 20<sup>th</sup> Intl. Conf. on Very Large Data Bases, September 12–15, 1994, Santiago de Chile, Chile*, pages 487–499. Morgan Kaufmann, 1994.
2. Džeroski S. and Lavrač N., editors. *Relational Data Mining*. Springer-Verlag, Berlin, September 2001.
3. Forman G. Choose your words carefully: An empirical study of feature selection metrics for text classification. In *Proc. of the 6th European Conf. on Principles of Data Mining and Knowledge Discovery*, pages 150–162. Springer-Verlag, 2002.
4. John G. H., Kohavi R., and Pfleger K. Irrelevant features and the subset selection problem. In *Intl. Conf. on Machine Learning*, pages 121–129, 1994.
5. Kramer S., Lavrač N., and Flach P. Propositionalization approaches to relational data mining. In Džeroski S. and Lavrač N., editors, *Relational Data Mining*, pages 262–291. Springer-Verlag, September 2001.
6. Li W., Han J., and Pei J. CMAR: Accurate and efficient classification based on multiple class-association rules. In *ICDM*, pages 369–376, 2001.
7. Termier A., Rousset M.-C., and Sebag M. TreeFinder: a first step towards XML data mining. In *Proc. of the 2000 IEEE Intl. Conf. on Data Mining*, 2002.
8. Theobald M., Schenkel R., and Weikum G. Exploiting structure, annotation, and ontological knowledge for automatic classification of XML data. In *Intl. Workshop on the Web and Databases*, 2003.
9. World Wide Web Consortium. *Extensible Markup Language (XML) 1.0*, W3C Recommendation edition, 2000.
10. Zaki M. J. and Aggarwal C. C. Xrules: Effective structural classifier for XML data. In *Proc. of the 2001 IEEE Intl. Conf. on Data Mining*, 2001.

### Annotation:

#### *XML documents classification*

We present a new method for classifying XML documents that does not require any additional information like a XML scheme or DTD. A new method for transforming XML data into one table is introduced. We present results of experiments with data from Internet Movie Database (IMDb). We show that our method overcomes the previous work in terms of accuracy and  $F_1$  measure.