Rozšíření infrastruktury projektu Pikater

Specifikace softwarového projektu

Datum ukončení: září 2014

Vedoucí projektu: Mgr. Martin Pilát, Ph.D.

Řešitelé:

- Štěpán Balcar
- Jiří Smolík
- Jan Krajíček
- Peter Šípoš

Stručný popis projektu Pikater

V současnosti existující a průběžně vyvíjený projekt Pikater se snaží řešit problém tzv. meta-učení. Meta-učení reaguje na potřebu vybírat vhodné metody strojového učení pro aplikaci na danou datovou množinu. K tomu využívá znalosti různých vlastností datových množin (meta-dat) a také znalosti výsledků jednotlivých metod na ostatních datových množinách. Meta-data obsahují například počet vzorů v datové množině, počet atributů, typy atributů, jejich rozptyly a další.

Jedním z úkolů meta-učení je samotný výběr vhodné metody pro daná data – o tuto část se stará tzv. doporučovač, který na základě podobnosti meta-dat a znalosti předchozích výsledků na jiných datových množinách určuje vhodnou metodu pro novou, neznámou datovou množinu. Dalším úkolem je nastavení parametrů vybrané metody tak, aby její výsledky byly co nejlepší. K tomu slouží prohledávací algoritmus, který prochází různá nastavení parametrů a snaží se najít takové, které dá co možná nejlepší výsledky. Prohledávací algoritmy lze také kombinovat s doporučovači, které určí vhodné rozsahy parametrů k prohledávání.

Nastavování správných parametrů je důležité i při vytvoření nové metody strojového učení a jejího porovnávání s existujícími metodami. V takovém případě je nutné, aby metody byly porovnávány na základě co možná nejlepšího nastavení jejich parametrů, aby některá z nich nebyla znevýhodněna špatným nastavením parametrů.

Právě prohledávání parametrů je časově náročná operace, neboť vyžaduje relativně velký počet opakování trénování daného modelu strojového učení – a jedno trénování může trvat i několik desítek minut.

Současný stav

Systém Pikater je založen na paradigmatu multi-agentního programování a implementován v prostředí Java/JADE s využitím dalších knihoven (Weka, …). Jednotlivé metody jsou v rámci systému implementovány jako samostatní agenti, kteří si navzájem posílají zprávy. Agentů je několik typů, mezi nejdůležitější patří:

1. DataManager – stará se o komunikaci s databází – poskytování meta-dat, ukládání výsledků, atd.
2. MetadataQueen – počítá meta-data z nově zadaných datových množin.
3. ComputingAgent – stará se o vlastní výpočet modelu strojového učení, o jeho trénování.
4. Recommender (několik typů) – doporučuje vhodnou metodu na základě předchozích výsledků a znalosti meta-dat.
5. Search (několik typů) – prohledává prostor parametrů metod strojového učení a hledá pro danou metodu takové nastavení jejích parametrů, které povede k co nejlepším výsledkům.
6. Manager – stará se o všechny výpočty, které v systému probíhají – přijímá zadání nových výpočtů od uživatelského rozhraní, zpracovává je a předává dalším agentům. Po skončení výpočtu informuje uživatelské rozhraní o jeho výsledcích.
7. AgentManager – stará se o spouštění nových agentů a správu agentů v systému celkově.
8. GUIAgent – stará se o rozhraní systému a uživatele, zajišťuje načtení konfiguračního souboru a zobrazuje případné zprávy o průběhu výpočtu, které přicházejí ze systému, předává požadavky na výpočty agentovi Manager.
9. DataReader – načítá datové množiny ze souborů a poskytuje je pomocí FIPA-ACL zpráv ostatním agentům v systému.

Systém umí řešit všechny základní problémy meta-učení, jak byly popsány v předchozí části. Umí doporučovat vhodné metody a prohledávat parametry metod. Dále umí automaticky zjišťovat meta-data z datových množin.

Požadavky na nové funkce (rozcestník)

Cílem projektu je rozšířit infrastrukturu projektu Pikater o novou funkcionalitu. Konkrétně se jedná o:

1. Zrychlit výpočty rozplánováním na více strojů.
2. Zajistit optimálnější distribuci dat v rámci systému a přidat podporu pro nové formáty.
3. Zavést podporu práce více uživatelů včetně administrátorského rozhraní.
4. Vytvořit k systému přívětivé uživatelského rozhraní dostupné z webového prohlížeče.

Jednotlivé body budou dále rozvedeny v následujících kapitolách.

Plánování a distribuce výpočtů

V rámci systému probíhá několik různých druhů výpočtů – trénování modelů strojového učení, načítání a předzpracování dat. Tyto výpočty jsou časově náročné a z hlediska systému jsou dále nedělitelné. Na druhou stranu jich probíhá velké množství a mělo by je být možné distribuovat na více strojů.

Projekt tedy implementuje agenta (plánovače), který bude poskytovat službu spouštění výpočtů na různých strojích. Všechny experimenty, které budou v systému spouštěny, budou tuto službu využívat. Navržené řešení musí splňovat následující podmínky:

1. Schopnost reagovat na řádově různě dlouhé doby běhu – rychlé modely se trénují zlomky vteřin, pomalé i hodiny. Hrubý odhad doby běhu bude poskytnut uživatelem.
2. Šetrnost k využití síťové infrastruktury – omezení zbytečných přesunů dat mezi jednotlivými stroji, příp. agenty v rámci jednoho stroje (když už má agent data načtena, je lepší použít ho znovu na stejných datech, než data načítat znova v jiném agentovi).
3. Respektování priorit úloh daných uživatelem, příp. správcem systému. Možnost změny těchto priorit, případně úplné zrušení dříve zadané úlohy – již běžící výpočty z této úlohy mohou doběhnout.
4. Monitorování probíhajících výpočtů a sledování výpadku jednotlivých strojů. V případě výpadku stroje znovuspuštění výpočtů, které na něm běžely, na jiném stroji – není požadováno pokračování přerušeného výpočtu.

Distribuce a správa dat v rámci systému

V rámci systému se vyskytují (pro účely tohoto projektu) 3 hlavní typy dat:

- Soubory s datovými množinami.
- Kód implementující jednotlivé metody strojového učení
- Natrénované a uložené modely (agenti).

Datové množiny mohou být dost velké, řádově až stovky MB. V současnosti je problém distribuce dat řešen tak, že se v rámci systému posílají FIPA-ACL zprávy v ontologii, která přímo popisuje datovou množinu. Ačkoliv toto řešení umožňuje jednoduchou práci s daty v systému, zpracování těchto zpráv trvá dlouho a nedá se dobře použít ani pro relativně malé množiny. Úkolem projektu tedy bude distribuovat datové množiny:

- v jednotném formátu, kterému rozumí všichni výpočetní agenti,
- dostatečně efektivně na to, aby byla umožněna práce i s velkými množinami,
- co nejméně, ve smyslu opakovaného přenosu daných dat na jeden výpočetní uzel.

Systém v současnosti podporuje pouze data ve formátu ARFF (což je v zásadě CSV s informacemi o typu a jméně atributů a o cílovém atributu). Projekt přidá podporu pro zadání dat ve formátu XLS.

A nakonec bude úkolem projektu zajistit ukládání natrénovaných modelů (agentů) do databáze, za předpokladu takového nastavení uživatelem. Modely, které nebudou do určené doby (kterou půjde nastavit) označeny pro trvalé uložení, budou po uplynutí této doby automaticky smazány.

Podpora práce více uživatelů

Se systémem bude moci současně pracovat více uživatelů.

Systém bude rozlišovat mezi dvěma druhy uživatelů – administrátory a běžnými uživateli. Běžní uživatelé budou mít 2 druhy oprávnění: možnost přidání nové datové množiny nebo nové funkčnosti do systému. Administrátor může měnit oprávnění a povyšovat běžné uživatele na administrátory.

Každý uživatelé bude mít také přednastaven interval priorit. Každému svému výpočtu pak budou moci přiřadit prioritu v tomto intervalu. Jestliže tak neučiní, použije se výchozí priorita.

Webové rozhraní

Pro běžného uživatele

Aktuálně veškerá komunikace mezi systémem a uživatelem probíhá přes konzoli, resp. prostřednictvím konfiguračních souborů.

Nové uživatelské rozhraní bude realizováno jako webová stránka a bude splňovat následující požadavky:

1. Přihlašování.
2. Kontrola stavu běžících výpočtů uživatele – zobrazuje, které experimenty už jsou dokončené a které se právě zpracovávají. U složitějších experimentů ukazuje již dokončené části.
3. Možnost zadání nových výpočtů, a to pomocí „spojování krabiček – komponent“.
    a. Každá „krabička” představuje nějakou součást systému (zdroj dat, optimalizační algoritmus, dříve uložený natrénovaný agent, atd.), spojnice mezi krabičkami představují data tekoucí systémem (datové množiny).
    b. Místo metody strojového učení bude možné zadat speciální hodnotu (např. „?“), která znamená, že metodu má systém vybrat sám.

   c. Stejným způsobem se budou označovat i parametry, které má systém sám určit (prohledáváním nebo na základě doporučování).

   d. Při zadávání výpočtu může uživatel nastavit jeho prioritu a musí zadat řádový odhad doby, po jakou výpočet poběží. Tento odhad následně využije plánovač výpočtů.

4. Prohlížení výsledků experimentů uživatele. Možnost jejich exportu do CSV pro další zpracování mimo systém.
5. Vizualizace datových množin, jak vstupních souborů, tak výsledků ohodnocených jednotlivými metodami, a porovnání správného ohodnocení s ohodnocením poskytnutým metodou strojového učení.
6. Mazání výsledků experimentů ze systému – skryje zobrazení uživateli. Nastavení experimentu a jeho výsledky v systému zůstanou pro použití při meta-učení.
7. Možnost upozornění na dokončení výpočtu prostřednictvím e-mailu. Budou v něm zároveň stručné výsledky experimentu – výsledky metod na daných datech.
8. U každého trénování metody bude možné nastavit, zda se má trvale uložit celý natrénovaný model. V případě použití prohledávacích algoritmů bude zároveň možnost trvale uložit pouze nejlepší natrénovaný model.
   Trvale uložené agenty bude možné použít v dalších výpočtech bez jejich nového trénování.
9. Přidávání nových komponent -„krabiček“. „Krabička” musí být specializací jednoho z typů definovaných systémem (Search, výpočetní agent, doporučovač) a uživatel musí dodat implementaci v podobě JADE agenta, který umí na požádání poslat svoji konfiguraci (mj. pro účely zobrazení v GUI). Přidání úplně nového typu krabičky umožněno nebude. Implementace bude podléhat kontrole a schválení administrátorem systému, systém neposkytuje ochranu proti potenciálně škodlivé implementaci uživatelských agentů.
10. Přidávání nových datových množin do systému včetně nastavení jejich meta-dat (alespoň těch, které se nedají zjistit automaticky – který atribut odpovídá cílové třídě, popis obsahu souboru).
11. Zobrazení datových množin v systému a jejich meta-dat (bez editace).
12. Správa uložených natrénovaných agentů – zobrazení, mazání.

Webové rozhraní

Pro administrátora

Administrátor má následující možnosti:

1. Přidávat a odebírat uživatele.
2. Měnit oprávnění jednotlivých uživatelů včetně povyšování běžných uživatelů na administrátory.
3. Resetovat heslo uživatele.
4. Nastavovat priority jednotlivých uživatelů.
5. Měnit priority úloh nastavených nebo běžících v systému.

6. Mazat úlohy ze systému.
7. Schvalovat uživatelům přidávání nových „krabiček“ či datových množin, nebudou-li k tomu mít oprávnění.

Organizace projektu

Použité technologie

Projekt Pikater je napsán v jazyce Java s použitím infrastruktury pro vývoj multi-agentních systémů Jade a frameworku Spring. Metody strojového učení, které jsou zatím naimplementované, využívají systém Weka. Data jsou ukládána v databázi (buď embedded HSQLDB, anebo distribuované MySQL).

Projekt rozšíření infrastruktury Pikateru bude založen na:

- systému Jade ve smyslu spouštění výpočtů (předpokládá se využití již naprogramovaných agentů projektu Pikater, tedy také knihoven Spring a Weka),
- PostgreSQL databázi, která nahradí MySQL.
- Vaadin frameworku pro tvorbu uživatelského rozhraní,
- webovém serveru Apache Tomcat,
- vhodné JavaScriptové knihovně pro usnadnění tvorby grafického editoru experimentů a vizualizací dat.

Kontrola práce na projektu

Práce na projektu bude kontrolována na pravidelných schůzkách, které se uskuteční vždy minimálně jednou za měsíc. Zároveň bude projekt udržovaný ve veřejně dostupném gitovém repositáři tak, aby bylo kdykoliv možné zkontrolovat průběh prací.

Předběžná časová osa projektu

Do 31. 3. 2014 – Úprava ukládání a distribuce dat v systému, distribuce dat v systému, prototyp plánovače úloh. Prototyp uživatelského rozhraní.

Do 31. 6. 2014 – Dokončení plánovače, dokončení administrátorského rozhraní. Z velké části hotové uživatelské rozhraní s možností nastavit i složitější experimenty.

Do 31. 7. 2014 – Dokončení implementace.

Do 31. 8. 2014 – Odladěný systém a dokumentace.