

Rozdział 4

Wybór środowiska oprogramowania

Tomasz ROGALA, Paweł CHRZANOWSKI

4.1. Wstęp

Niniejszy rozdział zawiera informacje na temat wyboru środowiska oprogramowania do budowy i rozwoju szkieletowego systemu doradczego MMNET. Szczególną uwagę poświęcono językom wyższego rzędu opartym na programowaniu funkcjonalnym. Wśród nich wytypowano język Matlab oraz język R. Posiadają one zbliżoną składnię oraz filozofię tworzenia oprogramowania opartą na wykorzystaniu macierzowego zapisu danych. Z uwagi na przeznaczenie tych środowisk szczegółowo przeszukano zasoby funkcji tych języków umożliwiających budowanie i stosowanie sieci stwierżeń. W kolejnym kroku dla przydatnych funkcji umożliwiających budowanie i stosowanie sieci stwierżeń przeprowadzono odpowiednie testy oraz weryfikację. Po przeprowadzonych badaniach wybrano środowisko oprogramowania. Szczegółowe wnioski związane z wyborem środowiska zawarto w podrozdziale 4.3.

4.2. Zestawienie i weryfikacja dostępnych funkcji i procedur umożliwiających budowanie i stosowanie sieci stwierżeń

Wybór środowiska oprogramowania poprzedzono szczegółowym przeglądem ogólnodostępnych zasobów umożliwiających budowanie i stosowanie sieci probabilistycznych w środowisku Matlab i R. Niniejszy podrozdział zawiera informacje na temat weryfikacji działania wybranych bibliotek zebranych w trakcie realizacji zadania, których szczegółowy opis zawarto w raporcie [4.19], [4.20]. Podstawowym celem weryfikacji była ocena poprawności działania bibliotek oraz ocena ewentualnych ograniczeń ich stosowania w określonym kontekście np. ocena możliwości zastosowania odpowiedniego algorytmu wnioskowania dla sieci, w której zmienne reprezentowane są w postaci ciągłej i dyskretnej. Ze względu na różnorodne zagadnienia dotyczące m.in. składowania sieci, wizualizacji, rozwiązywania, testowania sieci itp., sprawdzanie poszczególnych funkcji i procedur realizowano osobno w ramach danego zagadnienia. W celu zapewnienia możliwości porównania sprawdzanych funkcji i procedur, w ramach tego samego zadania, starano się, aby było ono realizowane w oparciu o jednakowo przygotowane środowisko uruchomieniowe. Z uwagi na ograniczone możliwości wczytywania sieci zapisanych w powszechnie znanych formatach w różnych pakietach, w których stosowane są indywidualnie przyjęte formaty zapisu sieci, środowiska uruchomieniowe w ramach tego samego zagadnienia różniły się. W takim przypadku procedury i funkcje były najczęściej weryfikowane w oparciu o wybraną sieć zaimplementowaną w danej bibliotece.

Tam, gdzie było to możliwe działanie poszczególnych funkcji i procedur sprawdzano w oparciu o zbiór przykładów różnych typów sieci:

- małych (do kilku węzłów) dyskretnych sieci dwustanowych np.: `sprinkler.net`, `rats.net` [4.4],
- sieci dyskretnych o stosunkowo większej liczbie węzłów np.: `asia.net` (`chest_clinic`), `barley.net` [4.1],
- sieci hybrydowych z węzłami dyskretnymi i ciągłymi [4.1].

Szczegółowe uwagi dotyczące weryfikowanych funkcji i procedur zostały zamieszczone jako podsumowania poszczególnych zadań przez nie realizowanych. Uwagi ogólne, jak również informacje na temat licencji pakietów zamieszczono w końcowej części niniejszego raportu.

Budowanie i stosowanie tradycyjnych sieci stwierdzeń związane jest z koniecznością realizacji wielu zadań. Z uwagi na rozbudowany zakres zadania polegającego na wytypowaniu potencjalnych funkcji i procedur umożliwiających budowanie i stosowanie sieci stwierdzeń, zostało ono podzielone na szereg podzadań:

- składowanie sieci,
- transformacja formatów zapisu sieci stwierdzeń,
- edycja sieci stwierdzeń,
- wizualizacja sieci,
- rozwiązywanie sieci (w tym wyznaczaniem stanu węzłów swobodnych),
- trenowanie parametrów sieci,
- trenowanie struktury sieci,
- narzędzia analizy sieci,
- łączenie mniejszych sieci w większą sieć,
- zarządzanie stwierdzeniami w sieci stwierdzeń,
- inne procedury związane z budowaniem i stosowaniem sieci jak np.: generowanie przykładów i dynamiczne sieci przekonania.

4.2.1. Składowanie sieci

Weryfikacja dostępnych funkcji i procedur języka R i języka Matlab dotyczyła głównie możliwości zapisu, odczytu oraz analizy dostępnych formatów dla tych operacji powszechnie stosowanych dla sieci probabilistycznych. Weryfikowano kolejno procedury i funkcje języka R, a następnie Matlab. Dla każdego środowiska zestawiono osobne wnioski.

Język R:

readnet, savenet

Procedura umożliwia zapis i odczyt w formacie **.net* (format Hugin).

Bibliografia: Pakiet `deal` [4.4].

saveXML(toGXL())

Procedura umożliwia zapis i odczyt *tylko grafów* w formacie **.GXL*. Plik ten może być bezpośrednio transformowany do formatu XML.

Bibliografia: Pakiet `graph` [4.9].

loadHuginNet, saveHuginNet

Procedura umożliwia zapis i odczyt sieci w formacie `net`, który może być bezpośrednio zapisywany w formacie XML.

Bibliografia: Pakiet `graph` [4.8].

Podczas weryfikacji funkcji i procedur dotyczących składowania sieci sprawdzano czy możliwy jest zapis i odczyt sieci w tym również sieci hybrydowych.

Wnioski

- Podane funkcje **readnet**, **savenet** mogą być wykorzystane do zapisu i odczytu sieci utworzonych tylko w ramach pakietu [4.4] ze względu na własny format zapisu danych.
- Funkcje **readnet**, **savenet** nie działają prawidłowo z sieciami o formacie **.net (HUGIN)*. Zastosowanie funkcji **readnet**, **savenet** dla formatu **.net (HUGIN)* prawdopodobnie wymaga niewielkiej modyfikacji źródeł pakietu [4.4].
- Funkcja **saveXML(toGXL(gR))** umożliwia zapis struktury grafu w formacie GXL.
- Funkcje **loadHuginNet**, **saveHuginNet** prawidłowo realizują zapis sieci przekonań w formacie **.net (Hugin'a)* dla węzłów dyskretnych.
- Zapis i odczyt sieci hybrydowych przy użyciu funkcji **loadHuginNet**, **saveHuginNet** kończy się niepowodzeniem wynikającym z niezgodności rozmiarów tablic prawdopodobieństw warunkowych. W programie *Hugin* zapis węzłów dyskretnych odbywa się na dwa sposoby. Pierwszy z nich to opis prawdopodobieństw warunkowych za pomocą zbioru gaussowskich funkcji gęstości prawdopodobieństwa. Drugi sposób polega na dyskretyzacji węzła ciągłego i zapisania wartości CPT w węzle "interval node". W trakcie weryfikacji zbadano możliwość odczytu i zapisu sieci hybrydowych zapisanych zgodnie z pierwszym sposobem.

Matlab:

bif2bnt

Konwersja z formatu BIF do formatu BNT. Program umożliwia konwertowanie sieci zapisanych w formacie BIF do formatu stosowanego w pakiecie BNTtoolbox.

Bibliografia: Program bif2bnt [4.12]

Wnioski

- Konwerter bif2bnt działa prawidłowo. Konwerter ten udostępniany jest jedynie w postaci aplikacji on-lineowej na stronie [4.12].

4.2.2. Transformacja formatów zapisu sieci stwierdzeń

Stwierdzono brak gotowych funkcji i procedur do transformacji formatów zapisu sieci stwierdzeń w środowiskach Matlab oraz R. Możliwe jest opracowanie własnych procedur w oparciu o funkcje do składowania sieci opisanych powyżej w sekcji 4.2.1.

4.2.3. Edycja sieci stwierdzeń

Wyniki poszukiwań funkcji i procedur umożliwiających edycję sieci stwierdzeń zamieszczono poniżej.

Język R:

DAG

Funkcja do definiowania grafu skierowanego.

Bibliografia: Pakiet ggm [4.18].

adjMatrix

Funkcja transformacji macierzy krawędzi w macierz sąsiedztwa.

Bibliografia: Pakiet ggm [4.18].

allEdges

Funkcja wyszukuje zbiór krawędzi grafu.

Bibliografia: Pakiet ggm [4.18].

cliques

Funkcja znajduje kliki w grafie nieskierowanym.

Bibliografia: Pakiet ggm [4.18].

cmpGraph

Funkcja znajduje wzajemnie się uzupełniający graf w grafie nieskierowanym.

Bibliografia: Pakiet ggm [4.18].

insert

Wstawia połączenia pomiędzy węzłami.

Bibliografia: Pakiet deal [4.4].

network

Definiuje sieć Bayesowską.

Bibliografia: Pakiet deal [4.4].

node

Definiuje węzeł.

Bibliografia: Pakiet deal [4.4].

numbermixed

Zwraca liczbę możliwych sieci dla podanego zbioru węzłów.

Bibliografia: Pakiet deal [4.4].

prob

Definiuje prawdopodobieństwa warunkowe w węźle.

Bibliografia: Pakiet deal [4.4].

localprob

Zwraca wartość tablic CPT w węźle.

Bibliografia: Pakiet deal [4.4].

unique.networkfamily

Usuwa sieć już istniejącą ze zbioru.

Bibliografia: Pakiet deal [4.4].

query, tab, or

Funkcje służące do sieci przy bezpośrednim definiowaniu Tablic Prawdopodobieństw Warunkowych (CPT).

Bibliografia: Pakiet Grappa [4.17].

makeadj

Definiuje macierz sąsiedztwa określającą strukturę sieci.

Bibliografia: Pakiet Grappa [4.17].

vs

Określa zbiór wartości zmiennej.

Bibliografia: Pakiet Grappa [4.17].

fq

Pobiera obliczone wartości prawdopodobieństw dla węzła.

Bibliografia: Pakiet Grappa [4.17].

joint

Pobiera obliczone wartości prawdopodobieństw dla dwóch lub więcej zmiennych w węźle.
Bibliografia: Pakiet Grappa [4.17].

prop.evid

Wprowadza wartości do określonego węzła (jako *evidence*).
Bibliografia: Pakiet Grappa [4.17].

andtable

Generuje tablice prawdopodobieństw warunkowych na podstawie wyrażenia logicznego AND.
Bibliografia: Pakiet gRain [4.8].

ortable

Generuje tablice prawdopodobieństw warunkowych na podstawie wyrażenia logicznego OR.
Bibliografia: Pakiet gRain [4.8].

cptable

Definiuje tablice prawdopodobieństw warunkowych wg. $p(v|pa(v))$.
Bibliografia: Pakiet gRain [4.8].

compile

Weryfikuje niezależność sieci Bayesowskiej.
Bibliografia: Pakiet gRain [4.8].

compileCPT

Tworzy tablicę prawdopodobieństw warunkowych.
Bibliografia: Pakiet gRain [4.8].

querygrain

Zapytanie o niezależność sieci.
Bibliografia: Pakiet gRain [4.8].

grain

Definiuje sieć.
Bibliografia: Pakiet gRain [4.8].

Wnioski

- Pakiet ggn:[4.18]
 - brak możliwości definiowania Tablic Prawdopodobieństw Warunkowych (CPT),
 - pakiet ogranicza się jedynie do operacji na grafach reprezentowanych w postaci macierzy sąsiedztwa;
- Pakiet Grappa [4.17] umożliwia definiowanie i edycję sieci stwierżeń o węzłach dyskretnych.
- Pakiet gRain [4.8] umożliwia definiowanie i edycję sieci stwierżeń o węzłach dyskretnych.
- Pakiet deal [4.4] umożliwia definiowanie i edycję sieci stwierżeń o węzłach ciągłych i węzłach dyskretnych.
- Pakiety Grappa, gRain oraz deal umożliwiają edycję sieci stwierżeń, wybór jednego z nich zależy będzie od pozostałych funkcjonalności weryfikowanych i opisanych w ramach tego rozdziału.

Matlab:**dag**

Łączy węzły.

Bibliografia: Pakiet BNTtoolbox [4.3].

mk_bnet

Definiuje sieć.

Bibliografia: Pakiet BNTtoolbox [4.3].

tabular_CPD

Definiuje tablice gęstości prawdopodobieństw warunkowych (CPD) - dla węzłów ciągłych.

Bibliografia: Pakiet BNTtoolbox [4.3].

CPT

Definiuje tablice prawdopodobieństw warunkowych (CPT) - dla węzłów dyskretnych.

Bibliografia: Pakiet BNTtoolbox [4.3].

parents

Definiuje rodziców dla węzła.

Bibliografia: Pakiet BNTtoolbox [4.3].

Wnioski Biblioteka BNT umożliwia definiowanie i edycję sieci stwierdzeń o węzłach ciągłych i węzłach dyskretnych.

4.2.4. Wizualizacja sieci

Możliwość odpowiedniej wizualizacji grafu jest przydatnym elementem wspomagającym proces budowy sieci probabilistycznych, umożliwia łatwiejszą identyfikację relacji pomiędzy rozpatrywanymi zmiennymi. W związku z tym przeprowadzono analizę dostępnych funkcji w języku R i Matlab umożliwiających wizualizację grafów.

Język R:**drawnetwork**

Umożliwia użytkownikowi budowanie sieci przekonań na podstawie prostego interfejsu typu wskaż i kliknij. Nadaje się do wizualizacji mało złożonych sieci (od aut.).

Bibliografia: Pakiet deal [4.4].

genpicfile

Umożliwia wizualizację sieci z zastosowaniem grafiki wektorowej w formacie *pictex – graphs*.

Bibliografia: Pakiet deal [4.4].

plotmat

Wizualizacja prostych grafów na podstawie macierzy połączeń.

Bibliografia: Pakiet diagram [4.5].

giRaph

Wizualizacja grafów oraz ich edycja.

Bibliografia: Pakiet giRaph [4.7].

graph z zastosowaniem dodatkowego pakietu graphviz

Wizualizacja grafów oraz szereg funkcji związanych z budowaniem i edycją grafów.

Bibliografia: Pakiet graph [4.7].

plot.mathgraph

Wizualizacja prostych grafów w oknie programu R.

Bibliografia: Pakiet *mathgraph* [4.11].

plot.bn

Wizualizacja prostych grafów w oknie programu R.

Bibliografia: Pakiet *bnlearn* [4.2].

W trakcie weryfikacji działania funkcji i procedur dla potrzeb wizualizacji sieci sprawdzano możliwość wizualizacji struktury sieci zarówno dyskretnych jak i ciągłych.

Wnioski

- Pakiet *deal* umożliwia wyświetlanie jedynie prostych sieci bez wyróżnienia typów węzłów.
- Pakiet *diagram* posiada wiele opcji umożliwiających uzyskanie wizualnie atrakcyjnej postaci grafu, możliwa jest również wizualizacja grafów hybrydowych ale wymaga ona rozszerzenia polecenia *plotmat*.
- Pakiet *bnlearn* pozwala na prostą i czytelną wizualizację grafów (bez wyróżnienia typów węzłów).
- Pozostałe pakiety posiadają ubogie możliwości reprezentacji graficznej grafu.
- Procedura *genpicfile* nie nadaje się do bezpośredniego wykorzystania w celu opracowania oprogramowania MMNET, ponieważ wymaga użycia kompilatora TeX. Umożliwia wizualizację sieci z zastosowaniem grafiki wektorowej w formacie *pictex – graphs*. Zapis graficzny sieci przypomina składnię język LISP.

Matlab:**draw graph**

Wizualizacja niewielkich grafów.

Bibliografia: Pakiet *BNTtoolbox* [4.3].

adj2pajek2

Wizualizacja dużych struktur sieci w programie *pajek*.

Bibliografia: Program *pajek* [4.13].

W trakcie weryfikacji działania funkcji i procedur dla potrzeb wizualizacji sieci sprawdzano możliwość wizualizacji struktury sieci zarówno dyskretnych jak i ciągłych.

Wnioski

- Pakiet *BNT* pozwala na wizualizację tylko mało złożonych struktur grafów (bez wyróżnienia typów węzłów).
- Zastosowanie funkcji *adj2pajek* wymaga zastosowania aplikacji zewnętrznej w celu zilustrowania sieci.

4.2.5. Rozwiązywanie sieci

W trakcie weryfikacji funkcji i procedur obejmujących zagadnienia rozwiązywania (wnioskowania) sieci nie prowadzono szczegółowych testów dotyczących porównania wydajności oraz dokładności wyznaczania prawdopodobieństw posteriori dla algorytmów przybliżonych. Działania te nie zostały przeprowadzone dla języka *R* ze względu na to, że większość pakietów wykorzystuje ten sam algorytm wnioskowania (ang. minimum clique weight heuristic) znajdujący się w pakiecie *gRbase* oraz to, że niektóre z pakietów zawierają algorytmy wnioskowania w postaci

zewnątrznie dołączanych bibliotek, co może mieć istotne znaczenie przy ocenie wydajności tych algorytmów. Testowanie w języku *Matlab* wydajności jak również dokładności wyznaczania prawdopodobieństw posteriori dla algorytmów przybliżonych zostało szczegółowo opisane w pozycji [4.16]. Biorąc powyższe pod uwagę sprawdzenie funkcji i procedur obejmujących zagadnienie rozwiązywania sieci ograniczono do weryfikacji ich działania oraz przedstawienia przykładów środowisk uruchomieniowych. Szczegółowe wyniki przedstawiono w raporcie [4.19], [4.20].

Język R:

compile.grain

Utworzenie drzewa złączeń oraz potencjałów dla poszczególnych węzłów (oraz klik) metodą triangulacji.

Bibliografia: Pakiet gRain [4.8].

propagate.grain

Poszukiwanie równowagi sieci dla drzewa utworzonego poleceniem `compile.grain`.

Bibliografia: Pakiet gRain [4.8].

compile

Utworzenie drzewa złączeń.

Bibliografia: Pakiet gRappa [4.10].

equil

Poszukiwanie równowagi sieci dla drzewa złączeń.

Bibliografia: Pakiet gRappa [4.10].

Wnioski

- Podane funkcje w języku *R* realizujące wnioskowanie działają prawidłowo.
- Większość funkcji dotyczących zasadniczego algorytmu triangulacji opartych jest na zastosowaniu algorytmu *mcwh* (ang. minimum clique weight heuristic).
- W przypadku pakietu *gRappa* algorytm wnioskowania dostępny jest w postaci biblioteki *gRappa.dll*.

Matlab:

jtree_inf_engine

Implementacja algorytmu drzewa złączeń.

Bibliografia: Pakiet BNTtoolbox [4.3].

var_elim_inf_engine

Implementacja algorytmu eliminacji zmiennych.

Bibliografia: Pakiet BNTtoolbox [4.3].

enumerative_inf_engine

Algorytm bazujący na przeszukiwaniu wyczerpującym - dokładny dla sieci dyskretnych.

Bibliografia: Pakiet BNTtoolbox [4.3].

pearl_inf_engine

Algorytm *Perla* - dla polidrzew.

Bibliografia: Pakiet BNTtoolbox [4.3].

quickscore_inf_engine

Algorytm *quickscore* dla sieci o strukturze QMR.

Bibliografia: Pakiet BNTtoolbox [4.3].

likelihood_weighting_inf_engine

Algorytm ważonej wiarygodności - algorytm przybliżony.

Bibliografia: Pakiet BNToolbox [4.3].

gibbs_sampling_inf_engine

Algorytm oparty na próbkowaniu Gibbs'a - algorytm przybliżony.

Bibliografia: Pakiet BNToolbox [4.3].

belprop_inf_engine

Loopy belief propagation - algorytm przybliżony.

Bibliografia: Pakiet BNToolbox [4.3].

Wnioski

- Podane funkcje w języku *Matlab* działają prawidłowo.
- Biblioteka dotycząca funkcji rozwiązywania sieci w pakiecie *BNT* jest obszerna i bogata oraz pozwala na zastosowanie różnych algorytmów, które powstawały na przestrzeni lat, gdzie rozwijano sieci przekonań, można znaleźć algorytmy dedykowane dla określonych typów sieci, jak również szereg algorytmów dokładnych i przybliżonych dedykowanych dla różnych typów sieci.
- Badania porównawcze funkcji wnioskowania w sieci przekonań zaimplementowanych w języku *Matlab* zostały przedstawione w [4.16]. Algorytmy testowano zarówno pod kątem wydajności jak również dokładności uzyskanych wyników.
- W przypadku pakietu *gRappa* algorytm wnioskowania dostępny jest w postaci biblioteki *gRappa.dll*.

4.2.6. Trenowanie struktury sieci

Poszukiwano funkcji i procedur umożliwiających trenowanie struktury sieci.

Język R:**gs**

Uczenie struktury z ograniczeniami algorytmem Grow-Shrink.

Bibliografia: Pakiet bnlearn [4.2].

iamb

Uczenie struktury z ograniczeniami algorytmem Incremental Association.

Bibliografia: Pakiet bnlearn [4.2].

fast.iamb

Uczenie struktury z ograniczeniami algorytmem Fast Incremental Association.

Bibliografia: Pakiet bnlearn [4.2].

hc

Uczenie struktury z uwzględnieniem dodatkowych algorytmów strojenia Hill-Climbing.

Bibliografia: Pakiet bnlearn [4.2].

tabu

Uczenie struktury. Zmodyfikowany algorytm Hill-Climbing.

Bibliografia: Pakiet bnlearn [4.2].

compare

Porównuje dwie sieci Bayesowskie.

Bibliografia: Pakiet bnlearn [4.2].

Matlab:**learn_struct_mcmc**

Uczenie struktury sieci Bayesowskich z użyciem metody MCMC (Markov Chain Monte Carlo) lub lokalnego przeszukiwania.

Bibliografia: Pakiet BNTtoolbox [4.3].

learn_struct_K2, learn_struct_pdag_pc), learn_struct_pdag_ic_star

Uczenie z ograniczeniami struktury sieci (IC/PC oraz IC*/FCI).

Bibliografia: Pakiet BNTtoolbox [4.3].

Wnioski

- Biblioteka *BNT* środowiska *Matlab* zawiera dużą grupę specjalistycznych funkcji przeznaczonych do uczenia struktury sieci.
- Biblioteka *bnlearn* środowiska *R* jest przeznaczona do uczenia struktury sieci bayesowskich.
- Każda z powyżej przedstawionych bibliotek pozwala na realizację zadania uczenia struktury sieci bayesowskich w oparciu o różnorodne algorytmy, a wybór środowiska obliczeniowego decyduje o zastosowaniu jednej z nich.
- Wybór procedury uczenia jest zależny od typu rozpatrywanej sieci oraz jej stopnia złożoności.

4.2.7. Trenowanie parametrów sieci

Poszukiwano funkcji i procedur pozwalających na uczenie parametrów już zdefiniowanych struktur sieci zarówno o węzłach dyskretnych, o węzłach ciągłych jak i struktur mieszanych.

Język R:**learn**

Uczenie parametrów sieci o węzłach ciągłych i/lub węzłach dyskretnych na podstawie danych wejściowych.

Bibliografia: Pakiet deal [4.4].

jointprior

Obliczenie łącznego prawdopodobieństwa w węźle.

Bibliografia: Pakiet deal [4.4].

Matlab:**learn_params_em**

Uczenie MLE/MAP parametrów sieci metodą EM.

Bibliografia: Pakiet BNTtoolbox [4.3].

bayes_update_params

Sekwencyjne uczenie parametrów sieci Bayesowskich.

Bibliografia: Pakiet BNTtoolbox [4.3].

Wnioski

- Biblioteka *BNT* środowiska *Matlab* zawiera dużą grupę specjalistycznych funkcji przeznaczonych do uczenia parametrów sieci bayesowskich.

- Biblioteka *deal* środowiska *R* jest przeznaczona do uczenia parametrów sieci bayesowskich zarówno o węzłach ciągłych i/lub dyskretnych.
- Każda z powyżej przedstawionych bibliotek pozwala na realizację zadania uczenia parametrów sieci bayesowskich w oparciu o różnorodne algorytmy, a wybór środowiska obliczeniowego decyduje o zastosowaniu jednej z nich.
- Wybór procedury uczenia jest zależny od typu rozpatrywanej sieci oraz jej stopnia złożoności.

4.2.8. Narzędzia do analizy sieci

W ramach niniejszego podzadania rozpatrywano tylko specyficzne narzędzia do analizy sieci przekonań. Pominięto typowe algorytmy z teorii grafów dotyczących ich opisu np. algorytm wyznaczania najdłuższej ścieżki w grafie.

Matlab:

calc_mpe, calc_mpe_bucket

Podaje największą wiarygodność węzła.

Bibliografia: Pakiet BNToolbox [4.3].

Wnioski

- Nie znaleziono wyspecjalizowanych narzędzi umożliwiających analizowanie sieci w środowisku *R*.
- Przedstawione powyżej narzędzia działają prawidłowo.
- Nie obejmują one zaawansowanych narzędzi dotyczących analizy kolizji, badania wrażliwości, oceny wartości (znaczenia) zawartych w sieci informacji ponieważ udostępniane nieodpłatnie pakiety nie zawierają takich narzędzi.

4.2.9. Inne: Generowanie przykładów

Język *R*:

rnetwork

Umożliwia generowanie przykładów.

Bibliografia: Pakiet *deal* [4.4].

simulate.grain

Umożliwia generowanie przykładów.

Bibliografia: Pakiet *gRain* [4.8].

simulate

Umożliwia generowanie przykładów.

Bibliografia: Pakiet *gRappa* [4.10].

Wnioski

Wymienione funkcje działają prawidłowo. Podczas weryfikacji nie sprawdzano możliwości generowania danych niepełnych.

4.2.10. Inne: Dynamiczne sieci przekonań

W tym podrozdziale wskazano i zweryfikowano wstępnie działanie pakietów umożliwiających budowanie dynamicznych sieci przekonań. Z uwagi na rozbudowany zakres tego zagadnienia weryfikację ograniczono jedynie do przykładów środowisk uruchomieniowych pozwalających na budowę sieci dynamicznych.

Język R:

pakiet G1DBN

Pakiet pozwalający na budowę i stosowanie dynamicznych sieci bayesowskich.
Bibliografia: Pakiet G1DBN [4.6].

Matlab:

wybrane funkcje pakietu BNT

Umożliwiają budowanie, uczenie i stosowanie dynamicznych sieci bayesowskich.
Bibliografia: Pakiet BNTtoolbox [4.3].

Wnioski

Przedstawione powyżej narzędzia dla środowiska R oraz Matlab działają prawidłowo.

4.2.11. Podsumowanie

W niniejszym podrozdziale przedstawiono jedynie te funkcje i procedury, które zostały szczegółowo opisane w dokumentacji poszczególnych pakietów. Opisano funkcje i procedury, które są stosowane dla potrzeb sieci przekonań, jak również te, które są używane w teorii grafów z ograniczeniem do tych, które zdaniem autorów mogą okazać się przydatne podczas opracowywania oprogramowania. Szczegółowe wyniki dotyczące weryfikacji poszczególnych funkcji i procedur zamieszczono w [4.19], [4.20]. Raporty te obejmują również przykłady zastosowania powyższych funkcji.

Na podstawie analizy dostępnych narzędzi pod kątem rozpatrywanych w niniejszym raporcie zagadnień dla potrzeb stosowania sieci probabilistycznych stwierdzono, że:

- najwięcej funkcji i procedur jest dostępnych dla środowiska *R*,
- spora część dostępnych funkcji i procedur w języku *R* jest w fazie ciągłego rozwoju i wymaga dalszej szczegółowej analizy,
- nie znaleziono nowych pakietów w języku *Matlab*, które w znacznym stopniu rozszerzałyby możliwości powszechnie stosowanego pakietu *BNTtoolbox*,
- przebadane funkcje i procedury w większości działają poprawnie i wymagają ewentualnie niewielkich modyfikacji. Te, które nie działają zgodnie z opisem przedstawionym w instrukcji użytkownika pakietu zostały opatrzone stosownym komentarzem. Dotyczy to głównie funkcji i procedur rozwijanych w ramach języka *R*,
- zadania związane z łączeniem sieci w większą sieć nie są realizowane w weryfikowanych środowiskach,
- zarządzanie stwierdzeniami w sieci stwierżeń, może być jedynie realizowane na poziomie edycji treści stwierżeń w postaci nazw węzłów,

- dynamiczny rozwój środowiska *R* i jego pakietów powoduje, że funkcjonalność oraz jakość działania kolejnych wersji przedstawionych pakietów staje się coraz lepsza.

Poniżej zamieszczono informacje na temat licencji wykorzystywanych przez poszczególne pakiety:

- Pakiety *bnlearn*, *deal*, *diagram*, *g1dbn*, *gRain*, *igraph* mogą być wykorzystywane pod warunkiem zachowania licencji GNU GPL => 2.
- Pakiet *graph* może być wykorzystany pod warunkiem zachowania licencji *Artistic 2.0* co oznacza, że udostępniany pakiet może być kopiowany i rozpowszechniany w wersji pierwotnej, natomiast rozpowszechnianie zmodyfikowanej wersji jest tylko możliwe w przypadku jej wersji źródłowej wraz z wyraźnym wskazaniem tych elementów które zostały zmodyfikowane. Szczegóły licencji można znaleźć m.in. na <http://www.opensource.org/licenses/artistic-license-2.0.php>.
- *mathgraph* może być swobodnie rozpowszechniany i stosowany dla celów niekomercyjnych.
- *BNT* toolbox może być stosowany pod warunkiem zachowania warunków licencji GNU GPL.
- *gRappa* jest wolny od opłat dla edukacyjnych i niekomercyjnych celów badawczych.

4.3. Wybór języka programowania

Oprogramowanie Matlab oraz język *R* posiadają wiele wspólnych zalet umożliwiających rozwijanie oprogramowania MMNET w ramach tzw. programowania funkcjonalnego przy pomocy przygotowanych, ogólnie udostępnionych pakietów. Obydwa środowiska umożliwiają szybkie tworzenie oraz dużą łatwość odczytu i debugowania kodu tworzonego oprogramowania, w tym również możliwość opracowywania wydajnych funkcji i procedur napisanych w języku *C*.

Na podstawie przeglądu dostępnych funkcji i procedur umożliwiających budowanie i stosowanie sieci probabilistycznych stwierdzono, że zarówno Matlab jak i *R* umożliwiają budowanie modeli opartych na grafach skierowanych. W przypadku języka Matlab dominującym pakietem narzędzi, które mogą być w tym celu stosowany jest pakiet *BNT* (Bayesian Network Toolbox). W przypadku języka *R*, to zbiór pakietów *gRain*, *bnlearn*, *igraph* *gRBase* oraz inne. Rozbudowana funkcjonalność pakietu *BNT* języka Matlab dotyczy głównie grafów kierunkowych i nie posiada podobnych możliwości dla grafów nieskierowanych oraz łańcuchów. Ograniczone są również możliwości związane z interfejsem dotyczącym budowania grafów. Wśród pakietów języka *R* funkcjonalność dotycząca modeli opartych na grafach jest bardziej rozbudowana; począwszy od algorytmów z teorii grafów, po implementację grafów skierowanych, nieskierowanych i łańcuchów, jednakże i tutaj brak jest odpowiednio rozbudowanych reprezentacji graficznych grafów i ich zawartości.

Na podstawie analizy dostępności oraz weryfikacji działania funkcji i procedur umożliwiających budowanie i stosowanie sieci stwierdzono podjęto decyzję o wyborze języka programowania. Podczas wyboru wzięto pod uwagę również istotne różnice pomiędzy środowiskiem *R* i Matlab. Oprócz pozytywnych cech wspólnych dla obu środowisk występują istotne różnice. Do najważniejszych można zaliczyć:

- Język *R* w przeciwieństwie do języka Matlab jest nieodpłatnie rozpowszechnianym oprogramowaniem, jest rozbudowywany i rozpowszechniany na prawach licencji GPL-GNU, co umożliwia prowadzenie dalszych prac na zasadzie otwartego projektu.
- Język *R* jest skierowany głównie do społeczności zajmującej się modelami statystycznymi, do których należą również probabilistyczne modele graficzne.

Ostatecznie zdecydowano o wyborze języka R jako środowiska do budowy i rozwoju oprogramowania MMNET.

Należy podkreślić, że z uwagi na to, iż język R jest oprogramowaniem wolnym, ma on bardzo aktywną grupę użytkowników i twórców pakietów wykorzystujących to oprogramowanie zarówno dla celów zawodowych jak i naukowych. W obecnym kształcie język R posiada bardzo rozbudowaną funkcjonalność dotyczącą obliczeń statystycznych oraz w mniejszym stopniu wielu innych zagadnień z innych dziedzin takich jak: przetwarzanie sygnałów, bioinformatyka, analiza systemów, itd. Ponadto język R, podobnie jak Matlab, posiada również wiele pakietów umożliwiających integrację oprogramowania z innym oprogramowaniem (np. Excel), jak również wymiany danych z innymi systemami za pomocą formatu danych XML oraz przechowywania i manipulowania danymi zawartymi w bazach danych za pomocą języka SQL. Język R nie traci wiele na integralności poszczególnych pakietów z uwagi na to, że jest oprogramowaniem wolnym. Posiada wytyczone standardy tworzenia nowych pakietów oraz opracowywania dokumentacji, które rozpowszechniane są z pomocą repozytorium CRAN [4.14].

Można również zauważyć, że główne ośrodki naukowe zajmujące się rozwijaniem modeli graficznych są obecnie coraz bardziej zainteresowane integracją działań wielu ośrodków na rzecz utworzenia wspólnego pakietu umożliwiającego budowanie i stosowanie modeli graficznych. Można wymienić tutaj inicjatywę pakietu *gR* [4.15] w ramach języka R. Motywacją do wspólnych działań było w tym przypadku wiele opracowań wolnego oprogramowania dla modeli sieci, przy czym większość z nich posiadała ograniczoną funkcjonalność. Należy podkreślić, że biorąc pod uwagę rachunek zysków i strat również inne zespoły podjęły jak się wydaje słuszną decyzję dotyczącą rozwijania własnych zainteresowań naukowych w tej dziedzinie z zastosowaniem języka R.

Bibliografia

- [4.1] Strona internetowa firmy Hugin. Internet: <http://developer.hugin.com/documentation/samples/>, 03.2010.
- [4.2] Package bnlearn. Internet: <http://www.bnlearn.com/>, 11.2009.
- [4.3] Package bntoolbox. Internet: <http://people.cs.ubc.ca/~murphyk/Software/BNT>, 11.2009.
- [4.4] Package deal. Internet: <http://cran.r-project.org/web/packages/deal/index.html>, 11.2009.
- [4.5] Package diagram: Functions for visualising simple graphs (networks), plotting flow diagrams. Internet: <http://www.ci.tuwien.ac.at/gR/>, 11.2009.
- [4.6] Package g1dbn. Internet: <http://cran.r-project.org/web/packages/G1DBN/index.html>, 11.2009.
- [4.7] Package giraph. Internet: <http://www.math.aau.dk/~dethlef/giRaph>, 11.2009.
- [4.8] Package grain. Internet: <http://genetics.agrsci.dk/~sorenh/public/R/gRainweb/>, 11.2009.
- [4.9] Package graph. Internet: <http://www.ci.tuwien.ac.at/gR/>, 11.2009.
- [4.10] Package grappa. Internet: <http://www.stats.bris.ac.uk/~peter/Grappa/>, 11.2009.
- [4.11] Package mathgraph. Internet: <http://www.ci.tuwien.ac.at/gR/>, 11.2009.
- [4.12] Program bif2bnt. Internet: <http://www.digitas.harvard.edu/~ken/bif2bnt>, 11.2009.

-
- [4.13] Program pajek. Internet: <http://vlado.fmf.uni-lj.si/pub/networks/pajek/>, 11.2009.
- [4.14] Comprehensive repository network. Internet: <http://cran.r-project.org>, 11.2010.
- [4.15] Package for graphical models. Internet: <http://cran.r-project.org/gR>, 11.2010.
- [4.16] Chrzanowski P., Rogala T. Comparison of belief network inference algorithms. Burczyński T., Cholewa W., Moczulski W., redaktorzy, *Recent developments in artificial intelligence methods*, strony 35–42, Gliwice, 2007. Silesian University of Technology.
- [4.17] Green Peter J. *Grappa: R functions for probability propagation*, 2005.
- [4.18] Marchetti Giovanni M., Drton Mathias. *ggm: Graphical Gaussian Models*, 2006.
- [4.19] Rogala T. Biblioteka funkcji i procedur umożliwiających budowanie i stosowanie sieci stwierdzeń-czesc II. Raport wewnętrzny MMN-04-012, Politechnika Śląska, Katedra Podstaw Konstrukcji Maszyn, Gliwice, 2010. (wersja 1.10).
- [4.20] Rogala T. Weryfikacja i testowanie funkcji i procedur umożliwiających budowanie i stosowanie sieci stwierdzeń-czesc II. Raport wewnętrzny MMN-04-022, Politechnika Śląska, Katedra Podstaw Konstrukcji Maszyn, Gliwice, 2010. (wersja 1.10).