



Licenciatura em Sistemas e Tecnologias de Informação

Geração automática de Redes Bayesianas

**Desenvolvimento de protótipo para prova de conceito com a *framework*
*Infer.NET***

Projecto Final de Licenciatura

Elaborado por Carlos Mareco

Aluno nº 20101417

Orientado pelo Professor Joaquim Canhoto

Barcarena

Setembro 2013

Universidade Atlântica

Licenciatura em Sistemas e Tecnologias de Informação

Geração automática de Redes Bayesianas

**Desenvolvimento de protótipo para prova de conceito com a *framework*
*Infer.NET***

Projecto Final de Licenciatura

Elaborado por Carlos Mareco

Aluno nº 20101417

Orientado pelo Professor Joaquim Canhoto

Barcarena

Setembro 2013

O autor é o único responsável pelas ideias expressas neste documento

Agradecimentos

À minha família, em especial à minha mãe e irmã que incentivaram e apoiaram desde o início, e das mais diversas formas, o meu regresso aos estudos, agradeço do fundo do coração.

Ao Professor Joaquim Canhoto agradeço o desafio que me levou a desenvolver este trabalho, mas acima de tudo a colaboração, disponibilidade e dedicação demonstrados ao longo da sua execução.

Aos meus colegas, Marco, José, Paulo e Luís, pelo apoio, camaradagem e partilha de experiências e conhecimentos. Nestes 3 anos, deixámos de ser apenas colegas de curso e tornámo-nos bons amigos.

A todos os amigos que, de forma directa ou indirecta, me apoiaram e incentivaram ao longo destes 3 anos, mas em especial à Ana, à Joana e ao Pedro, que leram e releram este trabalho, deixo aqui o meu agradecimento.

Ao Professor Doutor Marcírio Chaves tenho de agradecer a determinação e nível de exigência enquanto docente de algumas unidades curriculares, sobretudo porque foi essa exigência que alargou os meus horizontes e viabilizou a minha continuidade no mundo académico.

Resumo

Geração automática de Redes Bayesianas - Desenvolvimento de protótipo para prova de conceito com a *framework* Infer.NET

Redes Bayesianas (RB) são diagramas que permitem organizar o conhecimento através da construção de relações causa-efeito e resultam da utilização das teorias relacionadas com a probabilidade condicionada.

Neste trabalho o objectivo geral era avaliar a *framework* infer.NET quanto à sua capacidade de gerar automaticamente uma Rede Bayesiana apenas com base nas variáveis do problema e nas respectivas tabelas de probabilidade.

Fundamentou-se o desenvolvimento deste conceito através da revisão bibliográfica que permitiu reflectir sobre as diferentes abordagens às Redes Bayesianas e aos algoritmos de melhoria, e ainda comprovar a actualidade e premência da utilização prática dos conceitos.

A modelação automática de RB e consequente avaliação de desempenho não foram testadas, no entanto a bibliografia sugere diferentes abordagens e metodologias que suportam esta teoria.

Os automatismos, e respectiva metodologia de avaliação de desempenho, com base em casos reais, utilizando, por exemplo, uma metodologia de aprendizagem supervisionada, apresentam-se como uma possibilidade para desenvolvimento futuro deste tema.

Palavras-chave: Redes Bayesianas, Algoritmos Genéticos, infer.NET

Abstract

Automatic Generation of Bayesian Networks - Development of prototype for proof of concept with the *framework* infer.NET

Bayesian Networks (BN) are diagrams that enable you to organize knowledge by building cause-and-effect relationships, resulting from the use of theories related to conditional probability.

In this project work the aim was to assess the *framework* infer.NET regarding their ability to automatically generate a Bayesian network, only based on the variables of the problem and on the relevant probability tables.

The development of this concept was supported through the review of references, which allowed reflecting on the different approaches towards Bayesian Networks and improved algorithms, proving the timeliness and urgency of practical use of the concepts.

Automatic modelling of RB and consequent performance evaluation have not been tested, however the bibliographic references suggest different approaches and methodologies that support this theory.

The automatic systems and their performance evaluation methodology, based on real cases, using, for example, a supervised learning methodology, present themselves as a possibility for a future development of this subject.

Keywords: Bayesian Networks, Genetic Algorithms, infer.NET

Índice

Agradecimentos	i
Resumo.....	iii
Abstract.....	iv
Índice	v
Índice de Figuras.....	vi
Índice de Tabelas.....	vi
1. Introdução	9
1.1. Motivação.....	9
1.2. Descrição da Área e Problemática	9
1.2.1. Redes Bayesianas	9
1.2.2. Algoritmos Genéticos	11
1.2.3. <i>Bayesian Optimization Algorithm</i>	13
1.3. Relevância do trabalho.....	16
1.4. Objectivo.....	17
2. Desenvolvimento.....	18
2.1. Revisão Bibliográfica.....	18
2.2. Metodologia	28
2.2.1. Programação probabilística.....	28
2.2.2. Programação probabilística em Infer.NET – um exemplo prático	30
2.2.2.1. Contexto	30
2.2.2.2. Variáveis aleatórias	30
2.2.2.3. Distribuição prévia (<i>prior</i>).....	32
2.2.2.4. Quantificar o problema: distribuições.....	33

2.2.2.4.1.	Distribuições condicionais	33
2.2.2.4.2.	Distribuições conjuntas	34
2.2.2.4.3.	Distribuições marginais	34
2.2.2.5.	Inferir uma posterior.....	36
2.2.3.	Uma abordagem mais sofisticada	36
2.2.3.1.	Modelos mais sofisticados	37
2.2.3.2.	Inferência mais sofisticada.....	38
2.2.3.3.	Como escolher o melhor modelo	38
2.2.4.	Programação probabilística com infer.NET	39
2.3.	Protótipo e resultados	39
2.3.1.	Protótipo.....	40
2.3.2.	Desenvolvimentos futuros	42
3.	Conclusão	44
4.	Bibliografia.....	45

Índice de Figuras

Figura 1 - Pseudocódigo para Bayesian Optimization Algorithm (BOA), adaptado de (Pelikan, Hierarchical Bayesian Optimization Algorithm, 2005).....	14
Figura 2 - Modelos gráficos	37
Figura 3 – Interface gráfico do protótipo	42

Índice de Tabelas

Tabela 1 - Distribuições condicionais	33
Tabela 2 – Distribuições conjuntas.....	34
Tabela 3 - Distribuição marginal para a variável Chuva.....	35
Tabela 4 - Marginal para a variável Chuva	35

Tabela 5 - Posterior para Nuvens36

1. Introdução

1.1. Motivação

São vários os autores que defendem a importância e o potencial de crescimento dos sistemas periciais e de apoio à decisão, nomeadamente nas áreas da saúde, quer no diagnóstico médico, quer nos sistemas de controlo e gestão (Amit X. Garg, et al., 2005) (Kawamoto, Houlihan, Balas, & Lobach, 2005) (Rainu Kaushal, Kaveh G. Shojania, & David W. Bates, 2003) (Roshanov, et al., 2013) (Tiwari, Tsapepas, Powell, & Martin, 2013).

Sendo a gestão de sistemas e infra-estruturas na área da Saúde precisamente, o centro da minha actividade profissional, aceitei o desafio de desenvolver um trabalho numa nova vertente; aproveitando conhecimentos adquiridos, desenvolvendo ideias e descobrindo caminhos.

A minha proposta é: desenvolver um protótipo que comprove a geração automática de Redes Bayesianas sem recurso a um perito humano e contribuir, assim, para a evolução dos sistemas periciais na área da saúde.

1.2. Descrição da Área e Problemática

1.2.1. Redes Bayesianas

O nome de “Redes Bayesianas” resulta da utilização das teorias relacionadas com a probabilidade condicionada, tendo sido estabelecido por Thomas Bayes (Bayes, 1764).

Redes Bayesianas (RB) são diagramas que permitem organizar o conhecimento através da construção de relações causa-efeito (Sucar, 2006).

Os sistemas de apoio à decisão, suportados por RB, são competentes por si só, ao gerar automaticamente predições ou ao aconselhar decisões, bem como em situações de inexistência de todas as evidências. Ou seja, as RB permitem processos de tomada de decisão na ausência da totalidade da informação (Pearl, 1988).

Segundo Martin Pelikan, uma rede Bayesiana é definida por dois componentes:

- **A estrutura**, que é codificada por um grafo direccionado acíclico, com os nós correspondentes às variáveis existentes no conjunto de dados modelado, e as extremidades correspondentes às dependências condicionais;
- **Os parâmetros**, que são representados por um conjunto de tabelas de probabilidade condicional onde para cada variável, dada qualquer instância das variáveis de que esta depende, é especificada uma probabilidade condicional.

Matematicamente, uma rede Bayesiana codifica uma distribuição conjunta de probabilidades dada por

$$p(X) = \prod_{i=1}^n p(X_i|\pi_i)$$

onde $X=(x_1, \dots, x_n)$ é um vector de todas as variáveis no problema; π_i é o conjunto de pais de X_i na rede (o conjunto de nós de onde existe uma extremidade para X_i); e $p(X_i|\pi_i)$ é a probabilidade condicional de X_i dados os seus pais π_i .

Inicialmente estes modelos eram construídos "à mão", baseados no conhecimento de um perito, mas nos últimos anos várias técnicas têm sido desenvolvidas para aprender a partir dos dados, tanto a estrutura como os parâmetros associados ao modelo. Também é possível combinar conhecimento especializado com os dados para complementar o modelo (Sucar, 2006).

Vários investigadores examinaram os métodos de aprendizagem de RB a partir de dados, incluindo Cooper e Herskovits (1991,1992), Buntine (1991), e Spiegelhalter (1993). Todos estes métodos têm os mesmos componentes básicos: uma métrica de pontuação e um procedimento de pesquisa. A métrica calcula uma pontuação que é proporcional à probabilidade posterior de uma estrutura de rede dos dados apresentados e do conhecimento prévio de um utilizador. O procedimento de busca gera redes para avaliação pela pontuação métrica. Estes métodos utilizam os dois componentes para identificar uma rede ou um conjunto de redes com altas probabilidades posteriores, e estas redes são então usadas para prever eventos futuros (Heckerman, Geiger, & Chickering, 1995).

Foi nos anos 90 que cresceu o interesse em determinar métodos de optimização que explicitamente modelassem as boas soluções encontradas até ao momento e usassem o modelo construído para orientar a busca seguinte. Esta linha de pesquisa em optimização estocástica foi fortemente motivada pelos resultados alcançados no campo da computação evolutiva. No entanto, a ligação entre estas duas áreas foi por vezes obscurecida. Além disso, a capacidade de construção de modelos muitas vezes não era suficientemente poderosa para resolver problemas de optimização mais complexos.

1.2.2. Algoritmos Genéticos

Ao longo das últimas décadas os algoritmos genéticos (AG) têm sido aplicados com sucesso a muitos problemas de negócios, engenharia e ciência. Devido à sua simplicidade operacional e ampla aplicabilidade, tornaram-se numa área cada vez mais importante da optimização computacional (Pelikan, Hierarchical Bayesian Optimization Algorithm, 2005).

Estes algoritmos são inspirados no princípio Darwiniano da evolução das espécies e na genética, e utilizam conceitos provenientes do princípio de selecção natural para abordar uma multiplicidade de problemas, em especial os relacionados com o processo de optimização. São algoritmos probabilísticos que fornecem um mecanismo de busca paralela e adaptativa, baseada no princípio de sobrevivência dos mais aptos e na reprodução. Robustos, genéricos e facilmente adaptáveis estes algoritmos consistem numa técnica amplamente estudada e utilizada em diversas áreas (Lucas, 2002).

Muito úteis nos casos em que o espaço de procura é muito grande, utilizam ideias como a herança, a mutação, a selecção natural e os cruzamentos. Os AG trabalham com uma população de *strings* (genoma), que são formas codificadas de soluções candidatas (chamadas indivíduos ou criaturas) para um problema de optimização, que com o progredir do algoritmo evoluem para se tornarem melhores soluções (Pacheco, 1999).

A maior parte da teoria de AG lida com os chamados “blocos de construção”. Entenda-se por blocos de construção todas as soluções parciais de um problema. O algoritmo genético manipula um grande número de blocos de construção de mecanismos de selecção e recombinação; reproduz e mistura blocos de construção.

Os AG mantêm uma população de candidatos à solução no domínio do problema, onde cada solução é representada por uma sequência de parâmetros e, para chegar a uma solução óptima ou quase óptima, aplica operadores de recombinação, selecção e mutação ao longo de repetidas gerações (Munetomo, Murao, & Akama, 2007).

O procedimento básico do AG, segundo Martin Pelikan, consiste nos seguintes passos:

- **Inicialização** - AG normalmente geram a primeira população de soluções de candidato aleatoriamente, de acordo com uma distribuição uniforme ao longo de todas as soluções possíveis;
- **Seleção** - Cada iteração começa por seleccionar um conjunto de soluções promissoras da população actual, baseada no desempenho de cada solução;
- **Variação** - Depois de seleccionar o conjunto de soluções promissoras, novas soluções candidatas são criadas, aplicando conceitos de recombinação (cruzamento) e mutação às soluções promissoras. Recombinação concilia subconjuntos de soluções promissoras pela troca de algumas das suas partes. A mutação perturba ligeiramente as soluções recombinadas para explorar a sua vizinhança imediata.
- **Substituição** - Depois de aplicar o cruzamento e mutação para o conjunto de soluções promissoras, a população de novas soluções candidatas substitui a original e a próxima iteração é executada (começando com a selecção), a menos que os critérios de rescisão sejam atingidos.

Em computação evolutiva, *Estimation of Distribution Algorithms* (EDA), que também são chamados de *Probabilistic Model-Building Genetic Algorithms* (PMBGA), são estudados no contexto da realização de pesquisas robustas e evolutivas sem o conhecimento à priori da ligação ao problema. Os EDA são algoritmos baseados na pesquisa da população e, portanto, semelhantes aos AG em muitos aspectos; no entanto, não empregam operadores genéticos explícitos, tais como cruzamentos e mutações. Ao invés, constroem modelos probabilísticos de uma colecção de soluções promissoras - com os valores mais elevados de aptidão (para problemas de

maximização) ou inferiores (para problemas de minimização) - e utilizam este modelo para criar a próxima geração.

Assim, a diferença entre PMBGA e AG tradicionais é a forma como esses algoritmos processam populações de soluções promissoras para gerar populações de novas soluções de candidatos. Em vez de aplicar cruzamentos aos pares de soluções seleccionados, para depois aplicar a mutação para cada uma das soluções resultantes, realizam-se as duas etapas seguintes:

- **Construção do modelo:** onde é construído um modelo probabilístico de soluções promissoras;
- **Modelo de amostragem:** onde o modelo construído serve de amostra para gerar novas soluções.

Os PMBGA diferem pela forma como lidam com as duas etapas acima descritas e como incluem mecanismos especiais de selecção e substituição para processar as populações de soluções.

1.2.3. *Bayesian Optimization Algorithm*

Um algoritmo baseado nos conceitos dos algoritmos genéticos, que usa a estimativa de uma distribuição de probabilidade de soluções promissoras (PMBGA), a fim de gerar soluções de novos candidatos, é proposto no final da década de 90 e dá pelo nome de *Bayesian Optimization Algorithm* (BOA) (Pelikan, Goldberg, & Cantú-Paz, BOA: The Bayesian Optimization Algorithm, 1999).

Semelhante a um AG, o BOA inicializa uma população de cadeias binárias geradas aleatoriamente e selecciona um subconjunto das soluções mais promissoras, com valores relativamente elevados de aptidão (para problemas de maximização) (Munetomo, Murao, & Akama, 2007).

O BOA constrói uma RB para a população de soluções promissoras e utiliza a rede construída para gerar novas soluções candidatas. Inicialmente, o BOA utilizava a métrica Bayesian-Dirichlet sujeita a uma restrição de modelo-complexidade máxima para discriminar os modelos concorrentes, mas outras métricas foram analisadas em trabalho posterior. Em todas as variantes do BOA, o modelo é construído por um algoritmo que iterativamente adiciona uma nova dependência que maximiza a sua qualidade. Outros operadores gráficos elementares – como remoções e reversões de

extremidades – podem ser incorporados, porém as adições de extremidades são as mais importantes. A construção é encerrada quando mais nenhuma melhoria é possível (Pelikan, Hierarchical Bayesian Optimization Algorithm, 2005).

```
t := 0;
generate initial population P(0);
while (not done) {
    select population of promising solutions S(t);
    build Bayesian network B(t) for S(t);
    sample B(t) to generate O(t);
    incorporate O(t) into P(t);
    t := t+1;
};
```

Figura 1 - Pseudocódigo para Bayesian Optimization Algorithm (BOA), adaptado de (Pelikan, Hierarchical Bayesian Optimization Algorithm, 2005)

O BOA gera a população inicial de sequências (*strings*) de forma aleatória com uma distribuição uniforme sobre todas as sequências possíveis. A população é actualizada para um número de iterações (gerações), cada uma composta por quatro etapas. Em primeiro lugar, as soluções promissoras são seleccionadas da população actual usando um método de selecção de AG, como a selecção por torneio ou truncamento. Em segundo lugar, é construída uma rede Bayesiana onde cabe a população de soluções promissoras. Tem terceiro, são geradas novas soluções candidatas por amostragem da rede Bayesiana construída. Em quarto lugar, as novas soluções candidatas são incorporadas à população original, substituindo total ou parcialmente a solução original. As quatro etapas acima são repetidas até que alguns critérios de rescisão sejam atingidos. Por exemplo, a geração pode ser finalizada quando a população converge para a singularidade, a população contém uma solução suficientemente boa, ou se chegou a um limite no número de iterações. O procedimento básico de BOA é descrito na Figura 1 - Pseudocódigo para Bayesian Optimization Algorithm (BOA), adaptado de .

Segundo Pelikan, para uma aplicação bem-sucedida de BOA, é necessário que o algoritmo seja capaz de aprender uma rede que reflecte as dependências e independências que decompõem o problema correctamente. Existem duas subáreas de aprendizagem de uma RB:

- **A estrutura de aprendizagem.** Primeiro, deve ser determinada a estrutura de uma rede, que depois define as dependências condicionais e as independências codificadas pela rede.
- **As probabilidades condicionais de aprendizagem.** A estrutura também identifica probabilidades condicionais que devem ser especificadas para um modelo completo. Depois de aprender a estrutura, os valores das probabilidades condicionais no que diz respeito à estrutura final devem ser aprendidos.

O Algoritmo de Optimização Bayesiana, proposto por Pelikan, é considerado um dos EDAs mais sofisticados. É um algoritmo que utiliza técnicas de modelação de dados por RB para estimar a distribuição conjunta de soluções promissoras e esta estimativa é utilizada para gerar novos candidatos a soluções.

Apesar da existência deste tipo de algoritmos de melhoramento e predição, fortemente baseados na incerteza, há também autores que defendem o perito humano. Wiegerinck, Kappen e Burgers, referem que a tecnologia da RB é apenas um lado do modelo. O outro é o conhecimento de domínio, que é talvez ainda mais importante para o próprio modelo. Portanto a modelagem de uma RB requer sempre uma estreita cooperação com especialistas de domínio. E mesmo assim, o modelo é, naturalmente, apenas um dos muitos componentes de uma aplicação, como a interface do utilizador, a gestão de dados ou a aceitação do utilizador, essenciais para tornar o aplicativo um sucesso (Wiegerinck, Kappen, & Burgers, 2010).

Em termos práticos estes conceitos/metodologias resolvem questões abrangentes nas mais diversas áreas, desde a saúde, à gestão, passando pelos sistemas de informação, às finanças, à engenharia e a tantas outras mais.

Podem-se destacar alguns trabalhos de investigação que recorrem de alguma forma a estes métodos, como por exemplo o trabalho acerca de epístase genética usando critérios Bayesianos, escrito por Xia Jiang e colaboradores em 2011, o trabalho de Nir Friedman e colaboradores apresentado em 2000 sobre a análise de expressões de dados usando RB, a aproximação Bayesiana por AG para análise de dados espectroscópicos e metabólicos escrita em 2011 por Elon Correa e Royston Goodacre ou a utilização de um algoritmo de optimização Bayesiana (BOA) para a gestão de horários de enfermeiros publicado em 2006 por Jingpeng Li e Uwe Aickelin,

ou, por último, a aplicação de RB ao modelo de risco do cancro da mama num relatório de John S. Uebersax publicado em 2004.

A relevância do tema denota-se ainda mais nos trabalhos recentes de investigação que levam ao desenvolvimento de novas métricas e metodologias que desenvolvem e melhoram os conceitos acerca de RB, BOA e AG. Por exemplo o trabalho de Harald Steck em 2012 sobre a aprendizagem da estrutura de RB, ou a construção a partir de regras de sistemas especialistas baseados em RB, estudo de Saravanan Thirumuruganathan em 2010, ou ainda o relatório técnico de Marcio Kassouf Crocomo e Alexandre Cláudio Botazzo Delbem sobre Optimização Bayesiana com Detecção de Comunidades editado em 2011.

1.3. Relevância do trabalho

Decisores humanos são muitas vezes confrontados com domínios altamente complexos, tendo de lidar com várias fontes de informação e várias fontes de incerteza. A qualidade da decisão é fortemente influenciada pela experiência prévia para interpretar correctamente os dados.

O apoio à decisão informatizada pode ajudar a melhorar a eficácia do decisor, aumentando a consciencialização e alertando o utilizador para situações incomuns que podem ter grande impacto.

As RB são amplamente aceites como uma metodologia de princípios para a modelagem de domínios complexos, com incerteza, em que diferentes fontes de informação devem ser combinadas, como necessário em sistemas inteligentes de apoio à decisão.

Com este trabalho pretende-se comprovar que através de um *software* protótipo, utilizando a *framework* infer.NET da Microsoft Research, se consegue gerar uma RB de forma automática apenas com base nas evidências obtidas de cada variável. Para segundo plano deixaremos a possibilidade de otimizar essa RB com base num algoritmo de optimização ou implementar mesmo um BOA.

1.4. Objectivo

É objectivo deste trabalho dar a conhecer as RB e os algoritmos de optimização, a sua evolução e conjugação na vertente prática de utilização em sistemas de apoio à decisão e sistemas periciais.

São objectivos específicos:

- Verificar a possibilidade de gerar uma RB, com diversas formas de inferência, utilizando a *framework* de investigação infer.NET, da Microsoft Research;
- Utilizando a mesma *framework*, verificar a possibilidade de gerar uma RB de forma automática apenas com base nos dados do problema;
- Verificar se é possível optimizar uma rede gerada automaticamente tendo por base um dos algoritmos de optimização estudados.

2. Desenvolvimento

2.1. Revisão Bibliográfica

Nesta revisão bibliográfica são apresentadas várias formas de abordagem às Redes Bayesianas e a diversos algoritmos de optimização, assim como várias aplicações práticas destes métodos em diferentes áreas. Focámo-nos em quatro artigos actuais sendo que um deles é uma análise de três aplicações práticas de RB.

Xia Jiang e Gregory F. Cooper propõem uma *framework* de rede Bayesiana para uma classe de sistemas de vigilância de espaço-tempo chamado BNST. No seu artigo “A Bayesian spatio-temporal method for disease outbreak detection” os autores assumem que um sistema que monitoriza uma região para um surto de doença é chamado de sistema de vigilância de surto de doença. Um sistema de vigilância espacial pesquisa padrões de surto da doença em sub-regiões espaciais da região monitorizada. Um sistema de vigilância temporal procura padrões emergentes de surto de doença, analisando como os padrões mudam em períodos de tempo recentes. Se um sistema não-espacial, não-temporal pudesse ser convertido para um sistema espaço-temporal, o desempenho desse sistema poderia ser melhorado em termos de detecção precoce, exactidão e fiabilidade. A *framework* proposta pelos autores é aplicada a um sistema não-espacial, não-temporal de detecção de surto de doença chamado PC, a fim de criar um sistema espaço-temporal chamado PCTS. Diferenças no desempenho de detecção de ambos os sistemas, PC e PCTS, são examinados. Os resultados mostram que a abordagem Bayesiana espaço-temporal tem um bom desempenho, em relação à abordagem não-espacial, não-temporal.

No artigo “A genetic algorithm-Bayesian network approach for the analysis of metabolomics and spectroscopic data: application to the rapid identification of Bacillus spores and classification of Bacillus species” Elon Correa e Royston Goodacre desenvolvem um novo algoritmo baseado em Redes Bayesianas e Algoritmos Genéticos que identifica e selecciona com precisão um subconjunto pequeno de chaves relevantes de espectros de massa (bio marcadores) para serem analisados. Uma vez identificados, este subconjunto de bio marcadores relevantes foi então usado com sucesso para identificar esporos de bacilos e identificar espécies de bacilos, através de um modelo de RB construído especificamente para este conjunto reduzido de recursos. A rápida identificação de esporos de bacilos e identificação bacteriana

são de fundamental importância devido às suas implicações nas intoxicações alimentares, a patogénese e a sua utilização como potenciais agentes de guerra biológica. Técnicas analíticas muito automatizadas, tais como a espectrometria de massa de pirólise (Py-MS), têm sido usadas para identificar esporos bacterianos. Este modelo compacto de classificação Bayesiana é parcimonioso, computacionalmente rápido para executar e a sua visualização gráfica permite a fácil interpretação das relações probabilísticas entre bio marcadores seleccionados. Além disso, comparam-se os recursos seleccionados pela abordagem de Rede Bayesiana de Algoritmo Genético (RB-AG) com as características seleccionadas por análise discriminante de mínimos quadrados parciais (PLS-DA). Os resultados de precisão de classificação mostram que o conjunto de recursos seleccionados pelo RB-AG é muito superior ao PLS-DA.

Jingpeng Li e Uwe Aickelin mostraram que os horários podem ser construídos imitando um agendador humano, utilizando um conjunto de regras que envolvem o conhecimento de domínio. Apresentam um Algoritmo de Optimização Bayesiana (BOA) para o problema dos horários de enfermeiros que escolhe as regras de agendamento apropriadas de um conjunto para a atribuição de cada enfermeiro. Baseado na ideia do uso de modelos probabilísticos, o BOA constrói uma rede Bayesiana para o conjunto de soluções promissoras e amostras destas redes para gerar novas soluções candidatas. Resultados computacionais de 52 instâncias de dados reais demonstram o sucesso desta abordagem. Recomendam também que o mecanismo de aprendizagem do algoritmo proposto pode ser adequado para outros problemas de agendamento. O seu trabalho pode ser lido no artigo "Bayesian Optimisation Algorithm for Nurse Scheduling".

Wim Wiegerinck e colaboradores analisam, no artigo "Bayesian Networks for Expert Systems, Theory and Practical Applications", três aplicações práticas de Redes Bayesianas focando-se apenas em modelos que são criados usando a experiência de domínio. Após uma breve revisão de modelos de abordagens comuns de modelação de RB, discutem em detalhe três aplicações de redes Bayesianas. Com esses aplicativos, pretendem ilustrar o poder de modelação e flexibilidade das redes Bayesianas que ultrapassa os aplicativos padrão. A primeira rede é aplicada a um sistema de apoio à decisão de diagnóstico médico. Uma característica distintiva desta rede é a grande quantidade de variáveis no modelo. O segundo envolve um aplicativo

de suporte à decisão petrofísica para determinar o teor de mineral num poço com base em medições do furo. Este modelo difere do padrão das RB pelas suas variáveis contínuas e relações não-lineares. Finalmente discute-se um aplicativo para identificação de vítimas por análise de parentesco, com base nos perfis de ADN. A característica distintiva deste aplicativo é que as RB são geradas e calculadas com base nas informações de cada caso.

Xia Jiang e Gregory F. Cooper desenvolveram um método para converter um sistema bayesiano de detecção de surtos não-espacial, não-temporal para um espaço-temporal. Usando essa estrutura, ampliaram o sistema denominado PC para criar um novo sistema espaço-temporal passando a chamar-lhe PCTS. Os resultados experimentais suportam que o PCTS fornece detecção de surtos de doença melhorados, em relação ao PC, da qual foi derivado. Além de muitas vezes detectar surtos mais cedo (numa determinada taxa de alertas falsos), o PCTS foi melhor em manter um sinal de detecção estável ao longo do tempo. O PCTS é um sistema específico de doente que modela cada indivíduo numa população. Tendo também a possibilidade desse sistema poder obter o melhor desempenho de detecção do que um que usa a hipótese, tendo sido utilizado para comparação o SaTScan-Mt. Sendo um sistema bayesiano e permitindo a modelação específica do doente, transmite várias vantagens para a *framework* BNST. Uma vez que é bayesiano, pode incorporar-se conhecimento especializado e conhecimento que é aprendido a partir dos dados num sistema BNST. Um método como SaTScan baseia a sua análise apenas em dados atuais e não é capaz de tirar proveito do conhecimento prévio.

No estudo de Elon Correa e Royston Goodacre foram analisados dados de um grupo diversificado de espécies de bacilos usando uma nova abordagem resultante da combinação da selecção variável do AG com a inferência probabilística de relacionamento da RB. Esta abordagem de fusão quimiométrica foi usada primeiramente para a classificação bem-sucedida de esporos versus biomassa vegetativa e posteriormente os mesmos dados usados para identificar as espécies de bacilos que estavam sob análise. Demonstra-se que o GA-BN foi capaz de descobrir novos bio marcadores de esporos e que estes foram validados pelas diferenças fisiológicas que ocorrem durante a esporulação.

Seleção variável é um aspecto importante de qualquer análise de dados multivariada que pretenda simplificar um conjunto de dados, reduzindo a sua

dimensionalidade e identificando as variáveis relevantes subjacentes sem sacrificar a precisão preditiva. Como resultado para a classificação de espécies o GA-BN reduziu significativamente a redundância na informação fornecida pelas variáveis realmente usadas para previsão.

O algoritmo GA-BN superou um método de classificação tradicional usado em quimiometria, nomeadamente PLS-DA, em todos os casos testados. Embora o GA-BN nem sempre tenha seleccionado o menor subconjunto de recursos, a precisão de classificação indica que sempre foram seleccionados os mais relevantes, em relação ao PLS-DA. As Redes Bayesianas exploram duas características principais do conjunto de dados de destino: associações entre variáveis e a força dessas associações. A saída gráfica do modelo GA-BN explicitamente informa sobre associações probabilísticas. Uma tabela de probabilidade condicional armazena a força das correlações dadas as associações exibidas sobre o modelo de gráfico. Conhecimento especializado e informações estatísticas facilmente podem ser introduzidos na RB, conforme demonstrado neste estudo. RB modelam a distribuição de probabilidade do domínio do problema e, portanto, pode-se calcular a distribuição de previsão sobre os resultados de possíveis saídas.

Jingpeng Li e Uwe Aickelin na sua pesquisa sobre a problemática dos horários de enfermeiros que se aborda no artigo, definem que o número de enfermeiros é fixo (aproximadamente 30 dependendo da instância de dados), e o objectivo é criar agendas semanais, atribuindo a cada enfermeiro um turno padrão da forma mais eficiente. O problema pode ser resolvido usando uma regra apropriada, de um conjunto de regras disponíveis, para atribuição a cada enfermeiro.

Agendadores humanos podem fornecer soluções de alta qualidade, mas a tarefa é tediosa e muitas vezes requer uma grande quantidade de tempo. Normalmente constroem agendas com base nas regras aprendidas durante a programação. Devido às limitações humanas, estas regras são geralmente simples. No entanto, horários gerados por humanos são de alta qualidade devido à capacidade do agendador para alternar entre as regras, baseando-se no estado actual da solução. Prevê-se que a solução proposta, baseada no BOA seja capaz de executar esta função.

No algoritmo BOA, a estrutura da rede Bayesiana pode ser fixa, (Pelikan et al 1999) ou variável (Muhlenbein e Mahnig 1999). No modelo proposto utiliza-se uma estrutura de rede fixa porque todas as variáveis são totalmente observadas. Na sua essência, a RB é um vector de tamanho fixo de regras e o objectivo da aprendizagem é encontrar os valores de variáveis de todos os nós para maximizar a probabilidade dos dados de treino contendo um número de casos independentes.

Com base na estimativa de probabilidade condicional, o algoritmo apresentado utiliza técnicas do campo da modelação de dados por redes Bayesianas para estimar a distribuição conjunta de soluções promissoras. Aos nós, ou variáveis, da rede Bayesiana correspondem os pares enfermeiro/regra individuais para que um cronograma seja construído passo a passo.

A probabilidade condicional de cada variável na RB é calculada de acordo com um conjunto inicial de soluções promissoras. Posteriormente, cada nova instância para cada variável é gerada utilizando as probabilidades condicionais correspondentes, até que todas as variáveis sejam geradas. Daí, no caso apresentado, ter sido obtida uma nova sequência de regra. Um outro conjunto de sequências de regras será gerado desta forma, alguns dos quais irão substituir sequências anteriores com base na selecção de aptidão. Se o critério de paragem não é atingido, as probabilidades condicionais de todos os nós da RB são actualizadas novamente usando o actual conjunto de sequências de regras promissoras.

Wim Wiegerinck e colaboradores na sua abordagem a três aplicações práticas de RB começam por afirmar que a modelação de sistemas inteligentes para aplicações do mundo real, inevitavelmente tem que lidar com a incerteza. Esta incerteza é devido à impossibilidade de modelar todas as diferentes condições e excepções que podem fundamentar um conjunto finito de observações.

Em modelos para aplicações do mundo real, o número de estados é tão grande que uma representação do modelo esparso é inevitável. Uma classe geral com uma representação que permite a modelagem com muitas variáveis são as RB, que estão hoje em dia bem estabelecidas como uma ferramenta de modelagem para sistemas especialistas em domínios com incerteza.

Inferência probabilística é o problema da computação das probabilidades posteriores das variáveis não observadas no modelo, tendo em conta as observações de outras variáveis desse mesmo modelo.

A especificação de uma rede Bayesiana pode ser descrita em duas partes: uma qualitativa e outra quantitativa. A parte qualitativa é a estrutura gráfica da rede. A parte quantitativa consiste da especificação das tabelas de probabilidade condicional ou distribuições. Idealmente, ambas as especificações são inferidas a partir dos dados. Sendo uma alternativa fazer a especificação de ambas as partes, à mão, em colaboração com especialistas de domínio.

Embora as redes criadas desta forma possam ser bastante complexas, o âmbito de aplicação destes pacotes de *software*, obviamente, tem as suas limitações.

Os autores discutem três modelos em que a abordagem padrão para modelação bayesiana descrita foi inviável por diferentes razões: o grande número de variáveis no primeiro modelo, a necessidade de modelar variáveis com valor contínuo no segundo modelo e a necessidade de criar modelos de dados dinâmicos no terceiro aplicativo.

O primeiro modelo foi desenvolvido para um aplicativo para suporte a decisões de diagnóstico médico (Promedas, em colaboração com a UMC Utrecht). A principal funcionalidade do aplicativo é listar as doenças mais prováveis dadas as evidências no doente (queixas, testes, exames físicos) que são inseridas.

O sistema destina-se a apoiar o diagnóstico em medicina interna geral, cobrindo um grande domínio médico com diversas especializações. No entanto, um nível considerável de detalhes em que as áreas de doença são modeladas é essencial para o sistema seja de uso prático. Para este aplicativo, isso significa que o modelo deve conter milhares de doenças e uma infinidade mais de relações entre doenças e evidências no doente. Com esses números de variáveis e relações, uma abordagem de modelação padrão é inviável.

O segundo modelo foi desenvolvido para um aplicativo de suporte à decisão de petrofísicos (em colaboração com a SHELL EP). A principal função desta aplicação é fornecer uma distribuição de probabilidade da composição mineral de um potencial reservatório com base em medições remotas de um furo. No modelo subjacente, o

número de variáveis é limitado. No entanto, as variáveis são valorizadas continuamente. Uma delas representa as fracções de volume de 13 minerais e portanto é uma variável contínua 13-D. Qualquer diferença, por sensível que seja, numa abordagem de rede Bayesiana padrão levaria a uma disrupção do espaço e do estado. Devido a restrições e constrangimentos não lineares, uma rede Bayesiana com distribuições lineares Gaussianas também não é uma solução.

Por fim, um aplicativo para identificação de vítimas por análise de parentesco, com base nos perfis de ADN (Bonaparte, em colaboração com a NFI). As vítimas devem ser correspondidas com pessoas desaparecidas numa genealogia de membros da família. Nesta aplicação, o modelo segue as leis Mendelianas de herança genética e os princípios de perfis de ADN. Inferência precisa de algum pré-processamento, mas é razoavelmente simples. Neste aplicativo, no entanto, o desafio é que a estrutura do modelo que depende da estrutura familiar da pessoa ausente. Essa estrutura será diferente de caso para caso, e uma abordagem padrão com uma rede estática é obviamente insuficiente. Nesta aplicação, a modelação é implementada no motor. O aplicativo gera RB dinâmicas com base nas informações do caso e em seguida, faz as inferências necessárias para as combinações.

Nos artigos apresentados, são utilizadas RB para modelação de um problema sendo comum a intenção de apresentar resultados melhorados com este tipo de método de predição.

Em cada análise são utilizadas abordagens diferentes, sendo utilizadas RB na sua essência explorando a experiencia de domínio, RB dinâmicas, aplicações de AG para optimização da RB ou utilizando um algoritmo de optimização Bayesiana (BOA).

A *framework* bayesiana abordada no artigo “A Bayesian spatio-temporal method for disease outbreak detection”, segundo os autores, tem um preço: para a usar correctamente é necessário dedicar um esforço considerável para obtenção e representação do conhecimento prévio. Esta crença de conhecimento pode mudar ao longo do tempo, de uma região para outra e assim ser difícil de gerir e manter. Por outro lado, um sistema orientado a dados, como SaTScan, facilmente pode ser aplicado em muitos contextos sem modificação. Assim, ambas as abordagens têm vantagens e desvantagens. No entanto, os autores acreditam que é possível melhorar

significativamente os métodos para adquirir, gerir e manter o conhecimento prévio sobre surtos de doenças dentro de um quadro de Redes Bayesianas.

Em conclusão, os investigadores do artigo “A genetic algorithm-Bayesian network approach for the analysis of metabolomics and spectroscopic data: application to the rapid identification of Bacillus spores and classification of Bacillus species”, desenvolveram um algoritmo baseado em Redes Bayesianas e em Algoritmos Genéticos e demonstraram a sua implementação num conjunto de dados bem descrito, compreendendo os espectros de massa de pirólise de uma grande variedade de diferentes espécies de bacilos analisadas tanto como células vegetativas e esporos. Pode afirmar-se que a estrutura de classificação hierárquica informada mostrou excelente identificação das diferentes espécies no estado esporulado, uma descoberta que, segundo o conhecimento dos autores, não foi mostrada antes para dados de Py-MS (espectrometria de massa).

O algoritmo BOA apresentado por Jingpeng Li e Uwe Aickelin para resolver a problemática dos horários de enfermeiros, ao contrário da maioria das abordagens baseadas em regras, tem a capacidade de construir agendas utilizando regras flexíveis, em vez de fixas. Resultados experimentais de problemas reais de criação de horários demonstraram a força da abordagem.

Embora a solução apresentada pelos autores seja baseada na problemática dos horários de enfermeiros, recomenda-se que a ideia principal do algoritmo BOA possa ser aplicada a muitos outros problemas de planeamento onde os horários serão construídos sistematicamente de acordo com regras específicas. Também se espera que a pesquisa dê algumas respostas preliminares sobre como incluir aprendizagem humana em programação de algoritmos e, portanto, pode ser de interesse para profissionais e investigadores nas áreas da computação evolucionária e agendamento.

No último artigo são analisadas três aplicações informáticas baseadas em RB para sistemas de apoio à decisão.

Promedas é um aplicativo para suporte à decisão de diagnóstico médico. O seu objectivo principal é encontrar um diagnóstico diferencial com base nos resultados de testes (anamnese, exame físico, testes de laboratório, etc.). Dado o grande número de variáveis, uma abordagem de rede Bayesiana convencional é inviável. Utilizou-se uma

abordagem de base de conhecimento em que a rede é compilada a partir de uma base de dados de relações fornecida por médicos especialistas. Para viabilizar a computação, projectou-se uma parametrização de modelo tratável.

O segundo é uma aplicação de RB a estudos petrofísicos para apoio à decisão. Os modelos de observação baseiam-se sobre a física dos instrumentos de medição. As variáveis físicas nesta aplicação têm um valor contínuo. Uma abordagem de rede Bayesiana ingénua com valores diferenciados falharia. Para manter o domínio contínuo é utilizado o algoritmo híbrido de Monte Carlo para inferência.

Bonaparte é uma aplicação de RB para identificação de vítimas por análise de parentesco, com base nos perfis de ADN. As RB são usadas para modelar as relações estatísticas entre os perfis de ADN de pessoas diferentes numa genealogia. Por inferência Bayesiana, são calculados rácios de probabilidade e probabilidade posterior das hipóteses, que são as quantidades de interesse para o investigador forense. As relações probabilísticas entre variáveis baseiam-se em princípios primordiais da genética. Uma característica desta aplicação é a derivação automática, dinâmica, de modelos a partir de dados, ou seja, a estrutura genológica da família de uma pessoa desaparecida.

Neste artigo foram descritas três RB para aplicações do mundo real. Esses modelos baseiam-se na mesma metodologia com RB padrão, mas vão além das limitações mencionadas. O modelo Promedas tem em várias ordens de magnitude mais variáveis. O modelo petrofísico tem variáveis contínuas com valor. O modelo de Bonaparte, assim como o modelo de Promedas têm relações não-estáticas.

Conclui-se com esta revisão bibliográfica, que as diferenças fundamentais destes seis modelos com RB são a abordagem do modelo de desenvolvimento e o poder operacional e flexibilidade das aplicações.

Discutindo em detalhe a aplicação de RB, nestas aplicações práticas, procura-se ilustrar o poder de modelação que ultrapassa os aplicativos padrão. Os domínios aplicativos dos modelos demonstram que as redes Bayesianas podem ser aplicadas numa ampla variedade de domínios com diferentes tipos de requisitos, de onde se salienta que a tecnologia de RB é apenas um lado do modelo.

Esta revisão bibliográfica contribuiu em muito para o trabalho a desenvolver, demonstrando a forte possibilidade de atingir o objectivo proposto e a importância da utilização destas metodologias por si ou em conjunto para o desenvolvimento de *software* aplicativo de apoio à decisão ou de sistemas periciais.

Da leitura efectuada retiramos estes principais pontos-chave:

- Melhor desempenho;
- Incorporar-se conhecimento especializado e conhecimento que é aprendido a partir dos dados;
- Nova abordagem resultante da combinação de AG com a inferência probabilística das RB;
- Redução significativa da redundância na informação fornecida pelas variáveis;
- Superam o método de classificação tradicional;
- Conhecimento especializado e informações estatísticas podem ser introduzidos nas RB;
- Modelação de sistemas inteligentes para aplicações do mundo real, inevitavelmente tem que lidar com a incerteza;
- Estrutura gráfica da rede e tabelas de probabilidade condicional ou distribuições, idealmente são inferidas a partir dos dados;
- Problemas reconhecidos em lidar com grande número de variáveis, necessidade de modelar variáveis com valor contínuo e a necessidade de criar modelos de dados dinâmicos;
- É possível melhorar significativamente os métodos para adquirir, gerir e manter o conhecimento prévio;

RB estão hoje em dia bem estabelecidas como uma ferramenta de modelagem para sistemas especialistas em domínios com incerteza, sendo conhecidos métodos e abordagens que eliminam ou ultrapassam alguns dos problemas encontrados em determinados estudos.

2.2. Metodologia

2.2.1. Programação probabilística

Os computadores são rigorosamente lógicos mas o mundo real não é, um facto simples que pode representar alguns verdadeiros desafios para programadores.

Tomando como exemplo a representação de uma palavra, através de um rabisco realizado num ecrã sensível ao toque. As pessoas geralmente não são muito cuidadosas ou consistentes da sua escrita, pelo que o rabisco pode corresponder a "bola", ou "bolo" ou talvez até mesmo "dado". O ser humano conseguirá certamente identificar o que foi escrito, mas do ponto de vista aplicativo, o valor da palavra correcta é fundamentalmente incerto. Porém, não é completamente incerto; "bola" é mais provável do que "galo" e podem descartar-se completamente as palavras como "jogador" ou "treino".

A problemática consiste em como representar essa incerteza num programa. Variáveis convencionais do tipo booleano ou inteiro têm valores bem definidos. Um rabisco precisa de uma variável, a que podemos chamar *qualquerRabisco*, e que representa qualquer uma das várias palavras possíveis mas que não se sabe ao certo qual. No entanto, essa variável deve representar a incerteza de uma forma que incorpora a compreensão da probabilidade que cada uma das palavras possíveis pode ser a correcta.

Programação probabilística é projectada para lidar com essa incerteza. Baseia-se em variáveis aleatórias, que são as extensões dos tipos de padrão que podem representar valores incertos. Cada variável aleatória representa um conjunto ou intervalo de valores possíveis e tem uma distribuição associada que atribui uma probabilidade para cada valor possível. A distribuição quantitativa representa a sua compreensão dos valores possíveis da variável e permite a utilização de análise estatística para compreender o comportamento da variável.

Assim *qualquerRabisco* é representada por uma variável aleatória. Surge, assim, a questão de como obter a distribuição da *qualquerRabisco*, as probabilidades para cada uma das suas palavras possíveis. Do ponto de vista do utilizador há uma relação de causa-efeito entre uma palavra e o rabisco associado. Com programação probabilística pode-se construir um modelo probabilístico, que define a forma como os

utilizadores transformam palavras em rabiscos. Este modelo reconhece, entre outras coisas, que a mesma palavra pode levar a diferentes rabiscos e que diferentes palavras podem levar a rabiscos semelhantes, e define as probabilidades associadas.

Como racionalizar a partir de um rabisco específico para o conjunto de palavras possíveis e suas probabilidades? A programação probabilística é baseada numa metodologia estatística conhecida como inferência bayesiana, que permite racionalizar a partir de uma observação — o rabisco — até à sua origem, as palavras possíveis e suas probabilidades com base num modelo probabilístico.

Um modelo geralmente tem um conjunto de parâmetros ajustáveis que regem o seu comportamento, então há que determinar as configurações desses parâmetros correctamente. Podem atribuir-se valores de parâmetro, com base na compreensão geral da escrita, o que pode funcionar razoavelmente bem. No entanto, a generalidade das pessoas escreve de modo diferente, então as palavras possíveis associadas a *qualquer Rabisco* e as suas probabilidades variam entre utilizadores. Para ajustar os parâmetros para o estilo de escrita particular de cada indivíduo, um programa probabilístico pode tratar esses parâmetros também como variáveis aleatórias e aprender os valores de parâmetro real baseados na interacção com o utilizador.

Os resultados são aparentemente bons, mas poderá subsistir a dúvida se não faltará alguma coisa. Um modelo mais sofisticado com mais variáveis poderia funcionar melhor ainda, se adicionarmos bastantes variáveis ao modelo, este poderia conter qualquer coisa. No entanto, geralmente atinge-se o ponto em que os rendimentos começam a ser decrescentes; de alguma forma a complexidade adicional começa a reduzir a qualidade do modelo. É preciso encontrar um equilíbrio entre precisão e complexidade. Esse equilíbrio é parte integrante da inferência Bayesiana, que inclui uma maneira eficiente para avaliar a qualidade do modelo chamada de evidência, e que pode ser utilizada para escolher o melhor modelo.

Isto pode parecer um bom princípio, mas depende de como realmente se vão implementar modelos e inferir as probabilidades. Geralmente existem várias maneiras de resolver problemas probabilísticos. No entanto a programação probabilística fornece uma forma natural e integrada para lidar com esta classe de problemas. A *framework* Microsoft® Infer.NET simplifica bastante o processo, fornecendo uma API de modelação, que simplifica a mecânica de construção de modelos probabilísticos.

Mesmo modelos complexos muitas vezes podem ser expressos com apenas algumas linhas de código; E um motor de inferência, que combina o modelo com as observações e manipula a matemática complexa de se inferir as probabilidades para valores possíveis da variável especificada.

É esta *framework* que nós propomos utilizar e que se detalha através de um exemplo apresentado no próximo capítulo.

2.2.2. Programação probabilística em Infer.NET – um exemplo prático

O cenário a seguir fornece uma maneira conveniente para ilustrar a programação probabilística de conceitos básicos e definir alguns conceitos e terminologia essencial.

2.2.2.1. Contexto

O Sr. Joaquim mora em Cascais e quando sai de casa pela manhã percebe que o relvado da casa está molhado.

Qual o motivo para o relvado se encontrar molhado? Terá sido porque choveu (C) ou porque o Sr. Joaquim se esqueceu de desligar o sistema de rega (S)?

A sua certeza aumenta sobre ambas as ocorrências porque o relvado está molhado.

Ao verificar que existem nuvens no céu, o Sr. Joaquim tem quase certeza de que choveu.

Inicialmente, quando nada se sabe, o motivo do relvado estar molhado é desconhecido. No entanto, observações e o raciocínio probabilístico podem ajudar a identificar o motivo e fornecer, ao processo, uma introdução aos princípios da programação probabilística.

2.2.2.2. Variáveis aleatórias

São necessárias diversas variáveis para representar o problema. Neste caso todas elas apenas assumem valores discretos, verdadeiro e falso, então a escolha óbvia é uma variável que pode tomar qualquer um dos dois valores possíveis. Uma do tipo booleano seria ajustada, no entanto, pode ser apenas verdadeira ou falsa, e não

havendo certeza sobre o valor real da variável é necessário observar todos os factos. Pode-se, no entanto, estimar a probabilidade de que cada uma das opções.

É necessária uma variável aleatória, que essencialmente estende os tipos de domínio padrão tais como *bool* ou *int* para manipular tanto valores deterministas como incertos.

Uma variável *aleatória* tem um conjunto ou intervalo de valores possíveis, que são extraídos do tipo de domínio. Por exemplo, os valores possíveis de uma variável aleatória *booleana* são verdadeiro e falso. Os valores possíveis de uma variável aleatória *inteira* são números inteiros sobre uma escala contínua, tais como 0, 1 ou $-\infty, \infty$.

Cada variável aleatória tem uma distribuição de probabilidade associada que especifica a probabilidade de cada valor possível.

Por exemplo, uma variável aleatória do tipo *booleano* poderia ter uma probabilidade de 70% de ser verdadeiro e 30% de ser falso.

Neste problema em concreto são necessárias 4 variáveis aleatórias (Nuvens, RelvadoMolhado, Rega, Chuva) que embora apenas assumam 2 valores (verdadeiro e falso) no protótipo desenvolvido previu-se que as variáveis possam assumir mais valores.

Mesmo sem poder definitivamente dizer por que motivo a relva está molhada, neste ponto, pode-se estimar a probabilidade de cada um dos motivos acontecer. Cada variável aleatória tem uma função associada, conhecida como uma distribuição de probabilidade — geralmente abreviado para apenas distribuição — que atribui uma probabilidade para cada um dos possíveis valores da variável.

Neste caso, as distribuições associadas com cada uma das variáveis são funções de distribuições discretas, que atribuem probabilidades a um conjunto enumerável de valores possíveis. Porque o valor real deve ser um dos valores possíveis, as probabilidades devem somar 100%.

2.2.2.3. Distribuição prévia (*prior*)

Em programação probabilística começa-se por definir uma distribuição inicial para cada variável aleatória, que é chamado uma distribuição prévia ou crença prévia, sendo comumente abreviado para apenas prévia ou *prior*. Uma *prior* define a compreensão (ou falta dela) que se tem de uma variável antes de todas as observações terem sido feitas. No caso do exemplo enunciado sobre o relvado do Sr. Joaquim, assumimos que nada sabemos sobre a probabilidade do céu se encontrar nublado, neste caso a probabilidade à priori para os valores de domínio será 50%.

A distribuição prévia é um ponto de partida útil, mas pode melhorar-se a compreensão, se forem observados os valores reais de uma ou mais das variáveis aleatórias do modelo. Nesse ponto, a variável deixa de ser incerta, o que permite fazer uma melhor estimativa das distribuições das outras variáveis.

No exemplo em estudo, depois de consultar a previsão meteorológica do dia anterior, pode-se inferir uma nova distribuição acerca das nuvens, que incorpora esta observação na distribuição prévia da variável Chuva e melhora a estimativa acerca do motivo do relvado estar molhado. Uma observação não suplanta a sua *prior* — essa informação é ainda válida — mas podem-se usar as informações adicionais para rever a crença prévia para que esta reflecta com precisão todos os dados disponíveis. A nova distribuição é conhecida como uma distribuição posterior, ou apenas posterior. É um ponto de partida útil, mas pode melhorar a compreensão, se se observar o valor real de uma ou mais das variáveis aleatórias do modelo. Nesse ponto, a variável não está incerta, o que permite que se faça uma melhor estimativa das distribuições das outras variáveis.

Na verdade, a distinção entre *prior* e posterior às vezes pode não ser muito clara. Uma maneira mais geral e mais útil de distinguir é:

- A *prior* representa a compreensão do sistema antes de fazer um determinado conjunto de observações.
- A posterior correspondente representa a compreensão do sistema depois de se terem feito as observações.

Supondo uma observação adicional: o relatório meteorológico sobre a pluviosidade dá uma estimativa de quando choveu naquela zona. Antes de receber este segundo relatório, a compreensão da situação é representada pela crença

posterior que teve depois de observar a previsão meteorológica. O posterior é, portanto, a escolha lógica para a nova *prior*, que pode em seguida ser usado para inferir uma segunda posterior. Este tipo de inferência é incremental na natureza e é referido como aprendizagem em tempo real. Em alternativa pode-se voltar à crença original e reconsiderar todas as evidências a partir do zero.

Quanto mais evidências existem menos importante é a crença prévia. Provas conclusivas devem substituir qualquer crença prévia, mas se não houver muitas evidências para trabalhar, a crença prévia tem muito mais valor.

2.2.2.4. Quantificar o problema: distribuições

A discussão anterior traz alguns argumentos gerais sobre como inferir a causa mais provável, mas é um pouco vaga sobre os números reais. Esta seção quantifica essa discussão.

2.2.2.4.1. Distribuições condicionais

Para inferir uma posterior, primeiro deve-se construir um modelo matemático para o cenário. Já se especificou uma prévia para as nuvens.

A maneira mais simples para começar a construção de um modelo é estimar a probabilidade dos eventos ocorrerem. Isso é conhecido como uma distribuição condicional, a distribuição da Chuva, condicionada por um determinado valor atribuído às Nuvens.

Para simplificar a discussão, especificaremos simplesmente as duas distribuições, como mostrado na Tabela 1.

		Chuva		
		S	N	
Nuvens	S	80%	20%	= 100%
	N	20%	80%	= 100%

Tabela 1 - Distribuições condicionais

Cada possibilidade para a ocorrência de nuvens influencia a queda de chuva, assim cada distribuição condicional somará 100%.

2.2.2.4.2. Distribuições conjuntas

Podemos utilizar as distribuições prévias e condicionais da seção anterior para construir um modelo que é conhecido como distribuição conjunta. Uma distribuição conjunta representa as probabilidades de todas as combinações possíveis dos valores possíveis da variável aleatória — chuva com nuvens, chuva sem nuvens, nuvens sem chuva e assim por diante. Esta distribuição contém a compreensão completa de todo o problema antes de fazer quaisquer observações.

		Chuva	
		S	N
Nuvens	S	40%	10%
	N	10%	40%

Tabela 2 – Distribuições conjuntas

Cada linha da Tabela 2 contém as probabilidades condicionais da seção anterior, multiplicadas pelas probabilidades prévias correspondentes à existência de nuvens. Porque a combinação entre a existência de nuvens e a queda de chuva deve ser uma dessas quatro possibilidades, o somatório é 100%.

Podem-se criar distribuições conjuntas de variadas maneiras. Todas elas produzem os mesmos números. A abordagem utilizada neste documento é, basicamente, causa-efeito — A existência de nuvens causa a queda de chuva.

Nuvens é a variável em que estamos interessados — e que queremos inferir uma posterior para então definirmos uma prévia para essa variável.

Chuva é a variável que podemos observar, então definimos distribuições condicionais para essa variável.

Causa-efeito é geralmente a maneira mais simples para construir um modelo probabilístico, mesmo se pretendemos racionalizar na direção inversa, como inferir a existência de nuvens com base no conhecimento da queda de chuva.

2.2.2.4.3. Distribuições marginais

Pode-se usar a distribuição conjunta para responder a uma variedade de perguntas. Suponha-se que queremos saber a probabilidade de queda de chuva.

Podemos calculá-la a partir da distribuição conjunta, somando a probabilidade associada à ocorrência de chuva quer haja ou não nuvens. Podemos fazer o mesmo cálculo para a ocorrência de nuvens, independentemente de chover ou não. A distribuição que surge da soma de uma variável na distribuição conjunta é chamada de distribuição marginal.

A Tabela 3 - Distribuição marginal para a variável Chuva mostra a marginal para a ocorrência de chuva. Antes de receber qualquer relatório ou consultar qualquer facto.

		Chuva	
		S	N
Nuvens	S	40%	10%
	N	10%	40%
		$\Sigma = 50\%$	$\Sigma = 50\%$

Tabela 3 - Distribuição marginal para a variável Chuva

Pode-se calcular a marginal para as Nuvens da mesma forma, que simplesmente retorna as probabilidades *prior* que foram especificadas anteriormente (Tabela 4).

		Chuva		
		S	N	
Nuvens	S	40%	10%	$\Sigma = 50\%$
	N	10%	40%	$\Sigma = 50\%$

Tabela 4 - Marginal para a variável Chuva

Até agora, as distribuições marginais basicamente dizem o que já se sabe. Tornar-se-ão mais interessantes, depois de fazermos uma observação e adicionarmos novas informações.

2.2.2.5. Inferir uma posterior

Ao consultarmos o relatório meteorológico de pluviosidade ocorrida, podemos usar essa observação para inferir uma posterior na variável Nuvens, que passa a conter uma estimativa melhorada da possibilidade de ter chovido.

A posterior é um marginal condicional, a marginal para Nuvens, condicionada pela observação de que na realidade choveu. Neste caso simples, obtém-se a posterior da tabela de distribuição conjunta.

A informação contida no relatório significa que podemos eliminar a possibilidade de não ter chovido da tabela, restando apenas os valores que indicam a probabilidade de haver nuvens sabendo que choveu. Normalizando novamente os valores contidos na tabela para que o total seja 100%, obtemos a distribuição posterior demonstrada na Tabela 5.

		Chuva		
		S	N	
Nuvens	S	40%	10%	= 80%
	N	10%	40%	= 20%

Tabela 5 - Posterior para Nuvens

O resultado não é completamente certo, mas tudo indica que o céu esteve nublado.

Na prática, os modelos são geralmente muito mais complicados do que o utilizado neste exemplo, o que por sua vez torna computação de posteriores muito mais complexa. Uma aplicação de programação probabilística para cenários do "mundo real" requer uma abordagem mais sofisticada para construção do modelo e inferência.

2.2.3. Uma abordagem mais sofisticada

A distribuição conjunta na seção anterior é um modelo muito simples, com apenas duas variáveis e um pequeno conjunto de valores possíveis. Um modelo mais sofisticado que poderia lidar com uma ampla gama de observações pode proporcionar

mais segurança. No entanto, a computação de posteriores para modelos mais sofisticados é correspondentemente mais complexa.

2.2.3.1. Modelos mais sofisticados

É possível lidar com as variáveis adicionais expandindo a tabela de distribuição conjunta na Figura 2. No entanto, para mais de duas variáveis, ou para variáveis com muitos valores possíveis, as tabelas rapidamente se tornam humanamente incontroláveis. Além disso, as tabelas são úteis apenas para distribuições discretas. Se pretendemos definir uma distribuição para a hora, ela deve representar uma escala contínua de valores, que nunca pode ser representada por uma tabela.

Uma abordagem mais flexível e poderosa para criar um modelo conceitual para a distribuição conjunta, é fazê-lo na forma de um gráfico que representa as relações entre as variáveis aleatórias do sistema. A

Figura 2 - Modelos gráficos mostra dois exemplos, onde (a) é o gráfico do exemplo estudado, e (b) estende esse gráfico para manipular variáveis aleatórias adicionais.

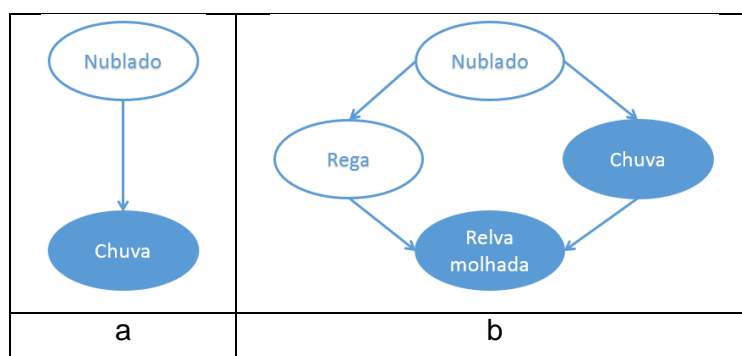


Figura 2 - Modelos gráficos

O modelo representa as relações entre variáveis aleatórias, como se segue:

- Cada caixa representa uma variável aleatória.
- As setas indicam as relações de causa-efeito entre variáveis aleatórias.
- As caixas sombreadas indicam as variáveis aleatórias que são observáveis.
- As caixas não sombreadas representam as variáveis aleatórias não observáveis e que gostaríamos de inferir.

No modelo (a) relaciona-se a existência de nuvens com a queda de chuva. Uma abordagem simplista do modelo completo (b) aqui utilizada para introduzir

conceitos. Podemos observar a queda de chuva para inferir a possibilidade de ocorrência de nuvens. Embora o modelo represente uma relação causa-efeito — que é conhecido como um modelo generativo — podemos usá-lo para racionalizar em qualquer direcção. Se soubermos o estado da nebulosidade, podemos usar essa observação para inferir a provável ocorrência de chuva.

Modelo (b) é um modelo mais sofisticado que incorpora variáveis aleatórias adicionais — a rega e o relvado molhado — onde o relvado molhado é também uma variável observável. Pode-se usar este modelo combinado com observações de várias variáveis aleatórias para determinar as posteriores, e talvez também produzir tal esmagadora evidência que torne o resultado indubitável.

2.2.3.2. Inferência mais sofisticada

Com este exemplo poderíamos usar aritmética simples para inferir uma posterior da tabela na Figura 2. A abordagem torna-se mais difícil à medida que adicionamos mais variáveis — especialmente se eles têm um grande número de valores possíveis — e não funciona para variáveis que representam uma faixa contínua de valores possíveis. Modelos mais realistas exigem uma forma mais sofisticada para inferir posteriores.

Os modelos conceituais na Figura 6 são na verdade uma representação gráfica da expressão matemática da distribuição conjunta. Programas probabilísticos podem usar essa expressão e a matemática da inferência Bayesiana para inferir as posteriores para grafos mais complexos, incluindo gráficos que possuem variáveis com distribuições contínuas. Podemos ainda usar a representação matemática para directamente definir modelos que não podem ser representados graficamente.

2.2.3.3. Como escolher o melhor modelo

O modelo (a) forneceu uma estimativa provável para a existência de nuvens ou a queda de chuva, mas talvez não seja suficientemente convincente. Um modelo mais complexo e sofisticado pode ser mais convincente. No entanto, adicionar mais variáveis não produz necessariamente um modelo melhor. Geralmente há um ponto de retorno decrescente, além do qual a complexidade adicional não acrescenta nada ao modelo, ou na verdade piora-o.

Por exemplo, quando se ajusta uma polinomial de um conjunto de pontos de dados, sempre se consegue um ajuste exacto, adicionando elementos suficientes para o polinómio. No entanto, um polinómio que se encaixa exactamente a cada ponto de dados normalmente balança descontroladamente entre cada ponto — um fenómeno conhecido como sobreajuste — que pode ser preciso mas não é muito útil. Um polinómio com menos elementos muitas vezes pode acondicionar bem os dados e fornecer um modelo muito mais útil e realista.

O que se pretende é um meio-termo: um modelo que encaixa os dados razoavelmente bem, sem ser excessivamente complexo. Assim, o melhor modelo é o mais simples que encaixe adequadamente os dados, dentro dos modelos possíveis.

Normalmente não existe certeza de qual o modelo ideal. No entanto, com a programação probabilística podemos tratar as evidências como variáveis aleatórias e inferir a distribuição dessas variáveis. A distribuição dá-nos a probabilidade em que cada modelo é o ideal, e pode-se usar essa informação para escolher o melhor modelo. Por exemplo, se determinada evidencia num modelo tem apenas uma probabilidade de apenas 15%, então essa variável adicional provavelmente apenas traz complexidade desnecessária para o modelo.

2.2.4. Programação probabilística com infer.NET

Programação probabilística é um conceito geral e pode ser implementada numa variedade de maneiras. A vantagem de usar Infer.NET, em suma, é a maneira simples que nos é oferecida de representar modelos gráficos em código e incluir um motor de inferência que manipula a matemática complexa de se inferir posteriores.

Pelas características apresentadas e pelos casos estudados, a *framework* de investigação da Microsoft Research infer.NET foi escolhida como ferramenta de trabalho exploratório deste trabalho.

2.3. Protótipo e resultados

Neste contexto foi desenvolvido um protótipo que utiliza a referida *framework*. As características diferenciadoras desta tecnologia estiveram presentes na sua escolha enquanto ferramenta de modelação. O Infer.NET permite modelar as redes de inferência em código, em tudo semelhante a C#, que após compilação gera automaticamente novo código em linguagem C# nativa. Este modelo gerado pode

posteriormente ser usado em inferências futuras, por exemplo em ambientes de produção. A separação resultante da compilação do modelo, e a possibilidade de o correr em separado é uma vantagem muito interessante, permitindo, por exemplo, refinar a capacidade de previsão do modelo face a novas evidências, sem ter de o reprogramar ou recompilar.

Um dos objectivos deste trabalho seria demonstrar a possibilidade de gerar RB automaticamente, recorrendo apenas às variáveis disponíveis e suas probabilidades independentes e condicionadas. A ideia assenta na possibilidade de construir automaticamente a rede (as relações entre as variáveis), avaliar o desempenho de cada configuração, e seleccionar a mais adequada.

O protótipo desenvolvido apenas aponta um dos caminhos possíveis na prossecução desse objectivo, demonstrando a viabilidade prática deste tipo de abordagem, utilizando as ferramentas já referidas. Está fora do âmbito do desenvolvimento deste protótipo encontrar uma solução para testar a eficácia de cada modelo de rede, no entanto a análise bibliográfica fundamenta esta abordagem.

2.3.1. Protótipo

Apresentam-se em seguida algumas das questões mais relevantes na utilização do infer.NET para a modelação de uma rede de inferência probabilística.

O primeiro passo é identificar as variáveis aleatórias (ou as variáveis disponíveis), que no nosso caso são:

- Nuvens,
- RelvaMolhada,
- Sprinkler,
- Chuva

No passo seguinte, utilizando já a API de desenvolvimento de software integrada com a *framework*, é necessário declarar as variáveis de acordo com as características de cada uma, adequando uma distribuição *prior* para cada uma dessas variáveis aleatórias. No caso de estudo as variáveis são todas discretas, o que limita o tipo de distribuição aplicável.

```
// variáveis aleatórias  
public VariableArray<int> Nuvens;
```

```
public VariableArray<int> Sprinkler;  
public VariableArray<int> Chuva;  
public VariableArray<int> RelvaMolhada;
```

As variáveis aleatórias representam os parâmetros das distribuições das variáveis aleatórias primárias. Para as variáveis “filhas” as probabilidades são declaradas na forma de tabelas de probabilidades condicionadas (CPTs).

```
public Variable<Vector> ProbCloudy;  
public VariableArray<Vector> CPTSprinkler;  
public VariableArray<Vector> CPTRain;  
public VariableArray<VariableArray<Vector>, Vector[][]> CPTWetGrass;
```

De seguida definem-se as probabilidades *prior*. Declarando variáveis, garantimos a possibilidade de fazer alterações sem ter de recompilar o modelo. As probabilidades *prior* são usadas como base de cada processo de inferência, e representam o conhecimento detido sobre o problema.

```
public Variable<Dirichlet> ProbCloudyPrior;  
public VariableArray<Dirichlet> CPTSprinklerPrior;  
public VariableArray<Dirichlet> CPTRainPrior;  
public VariableArray<VariableArray<Dirichlet>, Dirichlet[][]> CPTWetGrassPrior;
```

Declaram-se agora as variáveis que vão ser utilizadas para conter o resultado do processo de inferência.

```
public Dirichlet ProbCloudyPosterior;  
public Dirichlet[] CPTSprinklerPosterior;  
public Dirichlet[] CPTRainPosterior;  
public Dirichlet[][] CPTWetGrassPosterior;
```

Por último declaramos o motor de inferência.

```
public InferenceEngine Engine = new InferenceEngine();
```

No passo seguinte define-se a relação entre as variáveis (construção formal do modelo), as funções de aprendizagem e de *query* ao modelo.

A Figura 3 – *Interface gráfico do protótipo* apresenta o interface gráfico desenvolvido para testar o modelo, realizando inferência sobre a rede modelada.

Do lado esquerdo do interface pretende-se saber a probabilidade de ter chovido, com base nas CPTs das restantes variáveis do modelo. Pode-se modificar o conhecimento que se detém de cada uma das variáveis reflectindo-o na probabilidade final do evento ter ocorrido.

Do lado direito do interface procura-se saber a probabilidade da relva estar molhada, colocando o modelo a aprender com dados já conhecidos. Para a realização deste teste, são gerados aleatoriamente valores para as variáveis independentes com base nas distribuições conhecidas sobre essas variáveis. Este conhecimento poderia ser baseado num arquivo de dados registado numa base de dados.

Assim verifica-se que as probabilidades inferidas com base nos casos se aproximam das probabilidades conhecidas para aquelas variáveis, comprovando que a técnica de aprendizagem funciona correctamente.

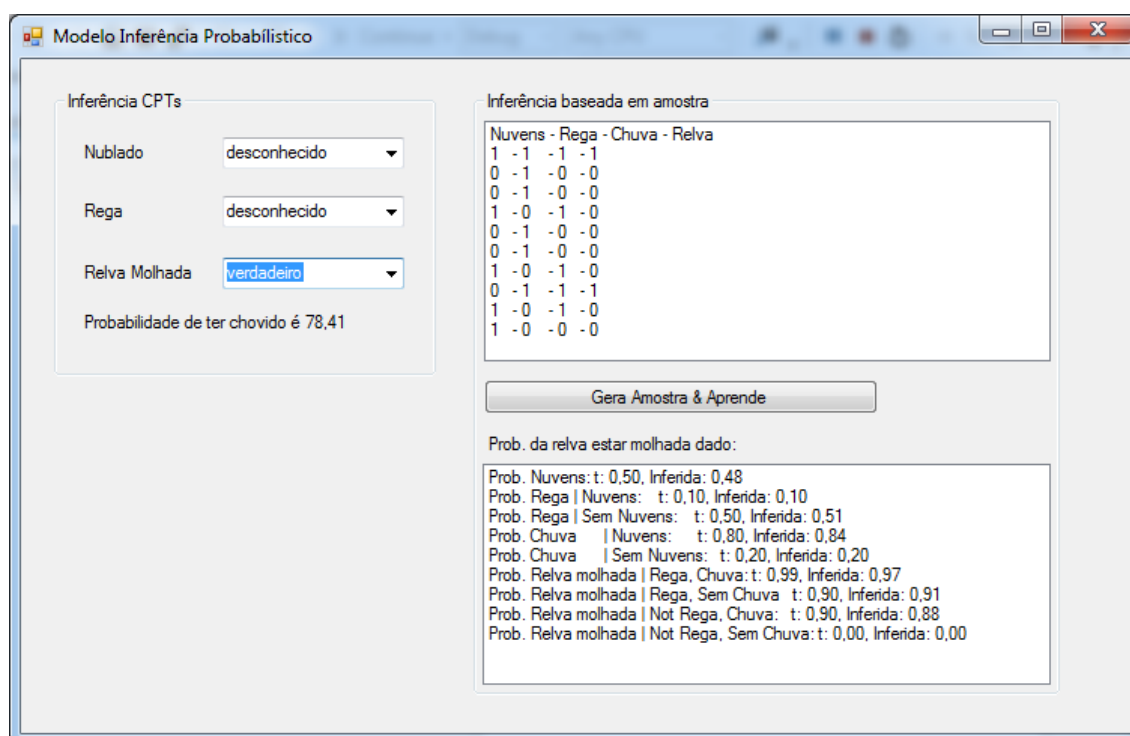


Figura 3 – Interface gráfico do protótipo

O desenvolvimento do protótipo permite alterar o modelo, no entanto, neste caso é necessário recompilá-lo.

2.3.2. Desenvolvimentos futuros

No protótipo desenvolvido a modelação automática de redes e consequente avaliação de desempenho não foi testada, no entanto parece possível utilizar esta API para atingir esse objectivo, sendo necessário aprofundar o conhecimento na linguagem de programação específica da *framework*.

Assim, os próximos passos passariam pelo desenvolvimento de um algoritmo de geração automática de RB e respectiva metodologia de avaliação de desempenho, com base em casos reais, utilizando, por exemplo, uma metodologia de aprendizagem supervisionada. A bibliografia sugere diferentes abordagens e metodologias que suportam esta teoria.

3. Conclusão

As Redes Bayesianas e os algoritmos de melhoria representam uma vasta área de estudo com inúmeras vertentes.

A abordagem seguida neste trabalho revela (quer pela revisão da literatura, quer pelo estudo prático desenvolvido) que se tratam de poderosas ferramentas de modelação de conhecimento, capazes de lidar com a incerteza e executar previsões variadas.

O estado da arte apresentado demonstra-nos que o tema é actual e que o uso prático é comum no desenvolvimento aplicacional de Sistemas de Apoio à Decisão e Sistemas Periciais.

Com o desenvolvimento do protótipo ficou demonstrado que é possível criar modelos de inferência Bayesiana com base na *framework* infer.NET.

Pese embora o objectivo proposto não tenha sido totalmente atingido, verifica-se que a ferramenta possibilita a construção de automatismos, seja com o objectivo de gerar automaticamente uma RB com diferentes configurações, mas também avaliar o seu desempenho recorrendo apenas a um conjunto de variáveis conhecido e respectivas probabilidades independentes e condicionadas.

No futuro, dando continuidade a este estudo, pretende-se desenvolver um algoritmo de geração automática de redes, e respectiva metodologia de avaliação de desempenho, com base em casos reais, utilizando, por exemplo, uma metodologia de aprendizagem supervisionada.

4. Bibliografia

- Amit X. Garg, M., Neill K. J. Adhikari, M., Heather McDonald, M., M. Patricia Rosas-Arellano, M. P., P. J. Devereaux, M., Joseph Beyene, P., . . . R. Brian Haynes, M. P. (2005). *Effects of Computerized Clinical Decision Support Systems on Practitioner Performance and Patient Outcomes*. American Medical Association.
- Bayes, T. (1764). An Essay towards solving a Problem in the Doctrine. Em *Philosophical Transactions of the Royal Society of London*.
- Correa, E., & Goodacre, R. (2011). *A genetic algorithm-Bayesian network approach for the analysis of metabolomics and spectroscopic data*. London: BMC Bioinformatics.
- Crocomo, M. K., & Delbem, A. B. (2011). *Otimização Bayesiana com Detecção de Comunidades*. São Carlos: Instituto de Ciências Matemáticas e de Computação.
- Friedman, N., Linial, M., Nachman, I., & Pe'Er, D. (2000). *Using Bayesian Networks to Analyze Expression Data*. Jerusalem: Journal of Computational Biology.
- Heckerman, D., Geiger, D., & Chickering, D. M. (1995). *Learning Bayesian Networks: The Combination of Knowledge and Statistical Data*. Seattle: Microsoft Research.
- Jiang, X., & Cooper, G. F. (2010). *A Bayesian spatio-temporal method for disease outbreak detection*. Pittsburgh: University of Pittsburgh.
- Jiang, X., Neapolitan, R. E., Barmada, M., & Visweswaran, S. (2011). *Learning genetic epistasis using Bayesian network scoring criteria*. London: BMC Bioinformatics.
- Kawamoto, K., Houlihan, C. A., Balas, E. A., & Lobach, D. F. (2005). *Improving clinical practice using clinical decision support systems*. BMJ Publishing Group.
- Li, J., & Aickelin, U. (2006). *BOA for Nurse Scheduling*. Illinois: Springer.

- Lucas, D. C. (2002). *Algoritmos Genéticos: uma Introdução*. Rio Grande do Sul: Universidade Federal do Rio Grande do Sul.
- Microsoft Research. (2011). *An Introduction to Infer.NET*. Microsoft Corporation.
- Microsoft Research. (2012). *Infer.NET 101*. Microsoft Corporation.
- Munetomo, M., Murao, N., & Akama, K. (2007). *Introducing Assignment Functions to Bayesian Optimization Algorithms*. Sapporo: Hokkaido University.
- Pacheco, M. A. (1999). *Algoritmos Genéticos: Princípios e Aplicações*. Universidade Católica do Rio de Janeiro.
- Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems*. San Francisco: Morgan Kaufmann.
- Pelikan, M. (2005). *Hierarchical Bayesian Optimization Algorithm*. Warsaw: Springer.
- Pelikan, M., Goldberg, D. E., & Cantú-Paz, E. (1999). *BOA: The Bayesian Optimization Algorithm*. Illinois: University of Illinois.
- Rainu Kaushal, M. M., Kaveh G. Shojania, M., & David W. Bates, M. M. (2003). *Effects of Computerized Physician Order Entry and Clinical Decision Support Systems on Medication Safety*. American Medical Association.
- Roshanov, P. S., Fernandes, N., Wilczynski, J. M., Hemens, B. J., You, J. J., Handler, S. M., . . . Haynes, R. B. (2013). *Features of effective computerised clinical decision support systems*. BMJ Publishing Group.
- Steck, H. (2012). *Learning the Bayesian Network Structure: Dirichlet Prior versus Data*. Malvern,: Computer-Aided Diagnosis (IKM CKS).
- Sucar, L. E. (2006). Redes Bayesianas. Em B. S. Araujo, *Aprendizaje Automático: conceptos básicos y avanzados* (pp. 77-100). Pearson Educación.
- Thirumuruganathan, S. (2010). *Building Bayesian Network based Expert Systems from rules*. Arlington: University of Texas.

Tiwari, R., Tsapepas, D. S., Powell, J. T., & Martin, S. T. (2013). *Enhancements in healthcare information technology systems*. JAMIA.

Uebersax, J. S. (2004). *Breast Cancer Risk Modeling*. RavenPack International SL.

Wiegerinck, W., Kappen, B., & Burgers, W. (2010). *Bayesian Networks for Expert Systems, Theory and Practical Applications*. Springer.