

NeticaTM

Application for Belief Networks
and Influence Diagrams

User's Guide

Netica Tutorial

https://www.norsys.com/tutorials/netica/nt_toc_A.htm

Netica Tutorial

https://www.norsys.com/tutorials/netica/nt_toc_A.htm

Kata Pengantar

Ingatlah saat melakukan tutorial bahwa ada banyak bantuan yang tersedia dari bantuan layar Netica, seringkali tentang network tutorial yang tepat. Bantuan Online adalah sumber terbaik untuk dokumentasi terbaru tentang fitur Netica. Untuk panduan yang disarankan, baca: Cara pendekatan tutorial ini

Catatan: link dalam daftar isi ini akan terbuka di tab baru, menggantikan dirinya sendiri saat topik lain diklik. Jika Anda mengklik link dan tidak muncul untuk membuka, periksa tab Tutorial lainnya yang terbuka di browser Anda.

Cara Pendekatan Tutorial Ini

Tutorial ini telah dirancang dengan tiga tipe pemikiran pembaca:

1. Orang awam yang cerdas secara teknis: Seseorang baru saja mulai dengan Bayes nets, yang tidak terbiasa dengan bagaimana mereka bekerja, atau apa yang dapat mereka lakukan, tapi siapa yang telah menggunakan komputer sebelumnya dan siapa yang mengetahui sesuatu tentang konsep probabilitasnya.
2. Insinyur: Insinyur atau ilmuwan yang memahami dasar-dasar probabilitas, yang telah mendengar tentang teknologi Bayes nets yang hebat, dan sedang mencari untuk belajar bagaimana menerapkannya di wilayahnya.
3. Pakar Bayes net: Pengguna net berpengalaman atau pakar Bayes, yang hanya ingin cepat mempercepat penggunaan Netica.

Kita merekomendasikan tiga tipe pendekatan untuk tutorial ini, tergantung pada kategori Anda.

Orang awam yang cerdas secara teknis

- Baca tutorial A1
- Tentukan apakah Anda cukup nyaman dengan konsep yang disajikan dalam tutorial itu, dan jika tidak, lakukan tinjauan singkat tentang teori probabilitas dasar sebelum melanjutkan.
- Download Aplikasi Netica dan instal di komputer Anda.
- Lanjutkan dengan sisa tutorial A, lalu lanjutkan B. Cadangan tutorial C dan D karena sampai Anda telah memiliki pengalaman aktual membangun Bayes nets dalam domain yang Anda minati.

Insinyur

- Skim tutorial A1
- Baca tutorial A2 sampai A4
- Tentukan apakah Bayes nets akan memecahkan kelas masalah Anda. Untuk membantu dalam hal ini, Anda mungkin ingin memeriksa Norsys net library untuk melihat apakah masalah serupa telah diselesaikan dengan menggunakan Bayes nets. Jika ya, lanjutkan.
- Download Aplikasi Netica dan instal di komputer Anda.
- Lanjutkan dengan sisa tutorial A, lalu B. Skim tutorial C dan D supaya Anda tahu apa saja yang tersedia.
- Mulailah membangun prototipe aplikasi Anda. Lihat tutorial C dan D sesuai kebutuhan.
- Jika Anda tahu bahasa pemrograman, sekarang Anda mungkin ingin membaca tentang Netica API toolkit yang tersedia. Panduan pengguna dan manual referensi mereka tersedia secara on-line.

Pakar Bayes net

- Download Netica dan Netica API.
- Lewati tutorial A.1, tapi skim sisa tutorial A untuk mengerti bagaimana hal-hal yang dilakukan dengan Netica.
- Baca tutorial B, yang akan menunjukkan dasar-dasar penting untuk segera bekerja dengan Netica.
- Lihat tutorial C dan D sesuai kebutuhan.

Daftar Isi

| | |
|---|-----------|
| A. Pengantar Bayes Nets | 2 |
| B. Operasi Netica Dasar | 21 |
| C. Intermediate Topics/ Topik Menengah | 37 |
| D. Advanced Topics/ Topik Lanjutan | 46 |

A. Pengantar Bayes Nets

1. Apa itu Bayes Net?

Mengapa berguna?

Mengapa disebut Bayes nets?

Untuk apa?

Diagnosis otomatis

Ramalan

Manajemen risiko keuangan, alokasi portofolio, asuransi

Pemodelan ekosistem

Fusi sensor

Monitoring dan peringatan

Interesting Properties dari Bayes Nets

2. Menggunakan Bayes net

Kompilasi net

Memasukan temuan/ findings

findings state yang positif

findings negatif

findings kemungkinan/ likelihood

Menghapus findings

3. Inferensi Probabilistik

Kompilasi, ditinjau ulang/ revisited

Menjelaskan/ explaining way

Pemblokiran/ blocking

4. Dasar Pengambilan Keputusan

Utility Node

Decision Node

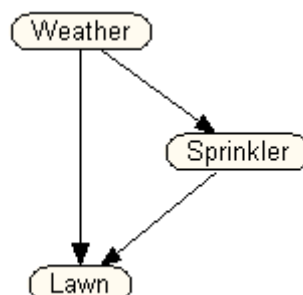
Teknik untuk mendapatkan valuations/ penilaian/ penaksiran yang bagus

1. Apakah Bayes Nets itu?

Bayes nets adalah model. Ini mencerminkan keadaan beberapa bagian dunia yang sedang dimodelkan dan ini menggambarkan bagaimana state-state tersebut terkait oleh probabilitas. Modelnya mungkin dari rumah Anda, atau mobil Anda, tubuh Anda, komunitas Anda, ekosistem, pasar saham, dll. Benar-benar segala sesuatu dapat dimodelkan oleh Bayes nets. Semua keadaan model yang mungkin mewakili semua kemungkinan dunia yang ada, yaitu, semua kemungkinan cara agar bagian atau state dapat dikonfigurasi. Mesin mobil bisa berjalan normal atau menimbulkan masalah. Ban itu bisa digelembungkan atau rata. Tubuh Anda bisa sakit atau sehat, dan sebagainya.

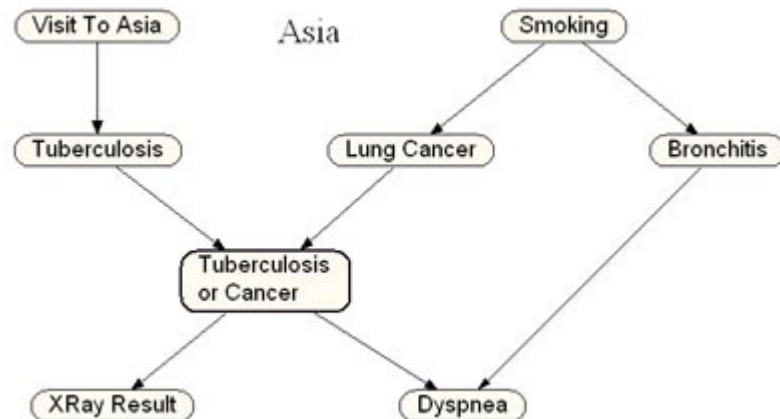
Jadi, dimana probabilitasnya datang? Nah, biasanya beberapa bagian state akan cenderung terjadi lebih sering ketika state lain hadir. Jadi, jika Anda sakit, kemungkinan pilek lebih tinggi. Jika mendung, kemungkinan hujan lebih tinggi, dan seterusnya.

Berikut adalah Bayes nets sederhana yang menggambarkan konsep-konsep ini. Didunia yang sederhana ini, mari kita katakan bahwa cuacanya bisa memiliki tiga keadaan: cerah, mendung, atau hujan, juga rumput bisa menjadi basah atau kering, dan penyiraman bisa menyala atau tidak. Sekarang ada beberapa hubungan kausal di dunia ini. Jika sudah hujan, maka rumput akan langsung basah. Tapi kalau sudah cerah untuk waktu yang lama, itu pun bisa membuat rumput basah, secara tidak langsung, dengan menyebabkan kita menyalakan sprinkler/ alat penyiram.



Bila probabilitas aktual dimasukkan ke dalam network ini yang mencerminkan realitas cuaca (wheater) yang nyata, rumput (lawn), dan perilaku penyiraman air (sprinkler), network semacam itu dapat dibuat untuk menjawab sejumlah pertanyaan bermanfaat, seperti, "jika rumput itu basah, apa kemungkinan itu disebabkan oleh hujan atau oleh sprinkler", dan "jika kemungkinan hujan meningkat, bagaimana hal itu mempengaruhi waktu saya untuk menyiram rumput?"

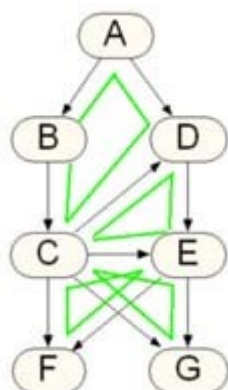
Berikut adalah Bayes nets sederhana lainnya yang disebut Asia. Ini adalah contoh yang populer untuk memperkenalkan Bayes nets dan berasal dari Lauritzen & Spiegelhalter⁸⁸. Perhatikan, ini hanya untuk tujuan, dan tidak boleh digunakan untuk pengambilan keputusan yang sebenarnya.



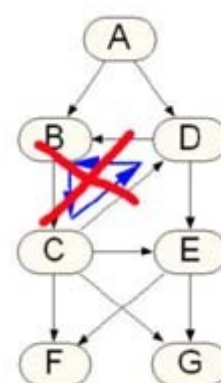
Ini adalah versi sederhana dari network yang dapat digunakan untuk mendiagnosa pasien yang tiba di klinik. Setiap node dalam network sesuai dengan beberapa kondisi pasien, misalnya, "Visit to Asia" menunjukkan apakah pasien tersebut baru saja mengunjungi Asia. Tanda panah (juga disebut link) antara dua node menunjukkan bahwa ada kemungkinan hubungan yang diketahui ada diantara keadaan kedua node tersebut. Dengan demikian, merokok meningkatkan kemungkinan terkena kanker paru-paru dan mendapatkan bronkitis. Baik kanker paru-paru maupun bronkitis meningkatkan kemungkinan terkena dyspnea (sesak nafas). Baik kanker paru-paru maupun tuberkulosis, tapi biasanya bukan bronkitis, bisa menyebabkan x-ray/ rontgen paru abnormal. Dan seterusnya.

Arah dari link panah kira-kira sesuai dengan "kausalitas/ causality". Itulah node yang lebih tinggi dalam diagram cenderung mempengaruhi yang dibawah ini daripada, atau, paling tidak, lebih daripada sebaliknya.

Dalam Bayes nets, link dapat membentuk loop, namun tidak membentuk siklus. Ini bukan batasan ekspresif; itu tidak membatasi kekuatan pemodelan network ini. Itu hanya berarti kita harus lebih berhati-hati dalam membangun network kita. Pada diagram kiri di bawah, ada banyak loop. Ini baik. Pada diagram yang kanan, penambahan link dari D ke B menciptakan sebuah siklus, yang tidak diizinkan.



Bayes nets yang valid



Bukan suatu Bayes nets

Keuntungan utama dari tidak membiarkan siklus itu memungkinkan update algoritma yang sangat cepat, karena tidak ada kemungkinan pengaruh probabilistik terhadap "siklus disekitarnya" tanpa batas waktu.

Untuk mendiagnosis pasien, nilai bisa dimasukkan untuk beberapa nodes saat diketahui. Ini akan memungkinkan kita untuk menghitung ulang probabilitas untuk semua node lainnya. Jadi jika kita mengambil x-ray dada dan x-ray adalah tidak normal/ abnormal, maka kemungkinan pasien yang terkena TB atau kanker paru-paru naik. Jika kita mempelajari lebih jauh bahwa pasien kita mengunjungi Asia, kemungkinan mereka menderita tuberkulosis akan meningkat lebih jauh, dan kanker paru-paru akan turun (karena sinar-X/ x-ray sekarang lebih baik dijelaskan dengan adanya TB daripada kanker paru-paru). Kita akan melihat bagaimana hal ini dilakukan dibagian selanjutnya.

Ringkasan

Pada bagian ini kita mengetahui bahwa network Bayesian adalah model, yang mewakili state-state yang mungkin ada di dunia nyata. Kita juga mengetahui bahwa Bayes nets memiliki hubungan probabilitas antara beberapa state di dunia nyata.

1.1. Mengapa Bayes Nets berguna?

1.1.1 Pemodelan kenyataan/ reality

Model umumnya berguna jika membantu kita untuk lebih memahami dunia bahwa kita adalah memodelkan, dan jika memungkinkan kita membuat prediksi berguna tentang bagaimana dunia akan berperilaku. Seringkali lebih mudah bereksperimen dengan model dibandingkan dengan kenyataan.

Dimasa lalu, ketika para ilmuwan, insinyur, dan ekonom ingin membangun model probabilistik dari dunia, sehingga mereka dapat mencoba untuk memprediksi apa yang mungkin terjadi ketika sesuatu terjadi, mereka biasanya mencoba untuk mewakili apa yang disebut "*joint distribution*/ distribusi bersama". Ini adalah tabel semua probabilitas dari semua kemungkinan kombinasi state dalam model dunia itu. Tabel seperti itu bisa menjadi besar, karena pada akhirnya menyimpan satu nilai probabilitas untuk setiap kombinasi keadaan, ini adalah perkalian dari semua jumlah state untuk setiap node.

Dalam model Cuaca/ Weather diatas, ini akan terjadi $3 \times 2 \times 2 = 12$ probabilitas.

Dalam model Asia memang begitu $2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 2^8 = 256$ probabilitas.

Untuk model dari kerumitan yang wajar, distribusi bersama bisa berakhir dengan jutaan, triliunan, atau banyak masukan yang luar biasa. Jelas cara yang lebih baik sangat dibutuhkan.

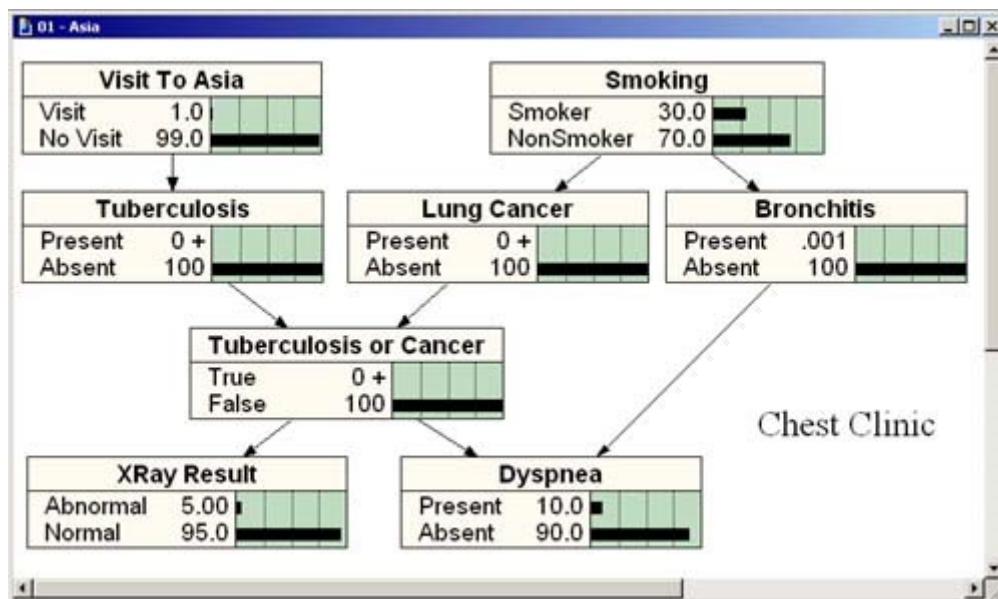
Bayesian nets adalah salah satu cara seperti itu. Karena Bayes nets hanya menghubungkan node yang terkait secara probabilistik dengan semacam ketergantungan kausal, penghematan besar dari perhitungan dapat terjadi. Tidak perlu menyimpan semua kemungkinan konfigurasi state, semua kemungkinan dunia, jika Anda mau. Semua yang dibutuhkan untuk menyimpan dan bekerja adalah semua kemungkinan kombinasi keadaan antara kumpulan node induk dan anak terkait (keluarga node, jika Anda mau). Hal ini membuat penghematan besar ruang tabel dan perhitungan. (Tentu saja, beberapa model masih terlalu besar untuk algoritma Bayes nets hari ini. Tetapi algoritma baru dikembangkan dan terobosannya menjanjikan, ini adalah area penelitian ilmu komputer modern yang sangat diteliti.)

Alasan kedua Bayesian nets terbukti sangat berguna adalah mereka sangat mudah beradaptasi. Anda bisa memulainya dari yang kecil, dengan pengetahuan terbatas tentang domain, dan menumbuhkannya saat Anda mendapatkan pengetahuan baru. Selanjutnya, saat Anda menerapkannya, Anda tidak memerlukan pengetahuan lengkap tentang instance dunia dari tempat Anda menerapkannya. Anda bisa menggunakan pengetahuan sebanyak yang tersedia dan net akan melakukan pekerjaan sebaik mungkin dengan pengetahuan yang ada.

Untuk menggambarkan hal ini, mari kita kembali ke Asia net kita, yang kita lihat di bagian 1 di atas. Mari kita anggap Anda adalah seorang dokter medis yang baru lulus di Los Angeles, spesialis penyakit paru-paru, dan Anda memutuskan untuk mendirikan klinik dada, yang menangani penyakit terkait paru yang serius. Dari studi buku teks Anda, Anda mengetahui sesuatu tentang tingkat kanker paru-paru, tuberkulosis, dan bronkitis, dan penyebab dan gejalanya, sehingga Anda dapat menyiapkan Bayes nets dasar dengan beberapa pengetahuan teoretis itu. Sebagai contoh, katakanlah sesuai dengan buku teks Anda:

- 30% populasi AS merokok.
- Kanker paru-paru dapat ditemukan pada sekitar 70 orang per 100.000.
- TBC terjadi pada sekitar 10 orang per 100.000.
- Bronchitis dapat ditemukan pada sekitar 800 orang per 100.000.
- Dyspnea dapat ditemukan pada sekitar 10% orang, namun sebagian besar disebabkan oleh asma dan penyebab selain TB, kanker paru-paru, atau bronkitis.

Berbekal statistik ini, Anda bisa membuat Bayes nets berikut ini:



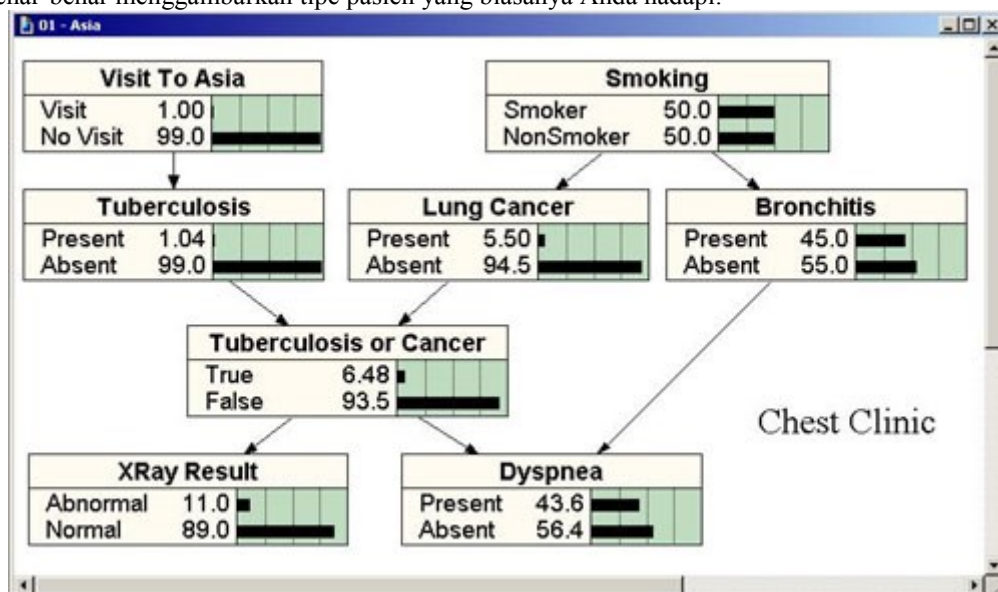
Sayangnya, network ini tidak terlalu membantu Anda, karena itu sama sekali tidak mencerminkan populasi orang yang mencari pertolongan dari klinik Anda. Kebanyakan dari mereka telah dirujuk oleh dokter keluarga mereka, sehingga kejadian penyakit paru di antara populasi tersebut jauh lebih tinggi, Anda bayangkan.

Jadi sebaiknya Anda tidak menggunakan Bayes nets diatas dalam latihan Anda. Anda membutuhkan lebih banyak data.

Saat klinik Anda tumbuh dan Anda menangani ratusan cases pasien, Anda belajar bahwa sementara buku teks mungkin menggambarkan situasi Amerika Utara, kenyataan klinik Anda dan populasi pasiennya sangat berbeda. Inilah adalah upaya pengumpulan data Anda terungkap:

- 50% pasien Anda merokok.
- 1% memiliki TB.
- 5,5% memiliki kanker paru-paru.
- 45% memiliki beberapa bentuk bronkitis ringan atau kronis.

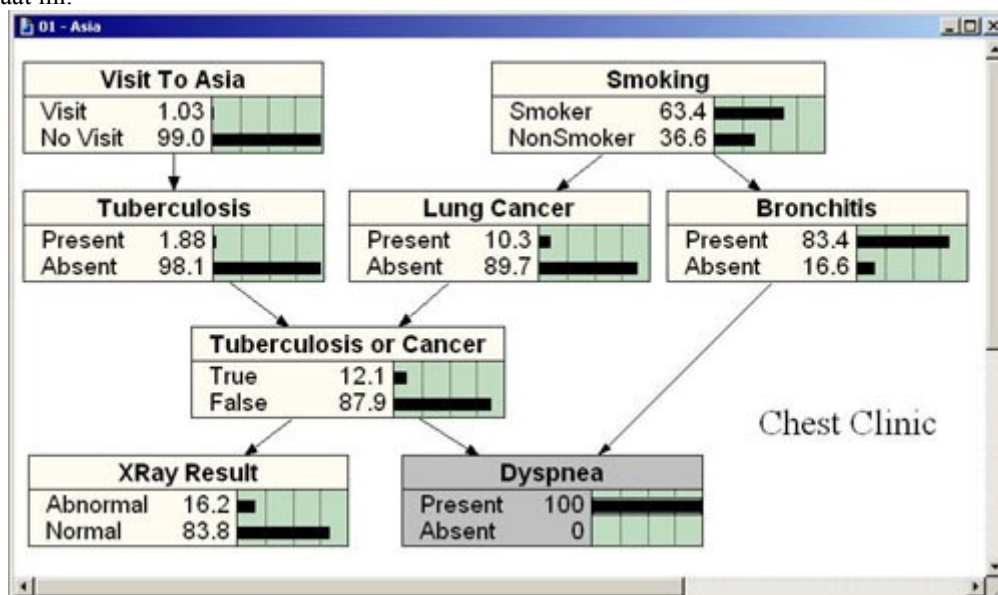
Anda memasukkan fitur baru ini ke dalam network Anda, dan sekarang Anda memiliki Bayes nets praktis, yang benar-benar menggambarkan tipe pasien yang biasanya Anda hadapi.



Jadi, mari kita lihat bagaimana kita akan menggunakan network ini dalam praktik medis sehari-hari kita.

Hal pertama yang harus kita perhatikan adalah bahwa menggambarkan net diatas seorang **new patient**, orang yang baru saja dirujuk kepada kita, dan untuk siapa kita **no knowledge**/ tidak ada pengetahuan apapun, selain itu mereka berasal dari populasi sasaran kita. Karena kita memperoleh pengetahuan khusus untuk setiap pasien tertentu, probabilitas di net akan secara otomatis menyesuaikan diri. Inilah keindahan dan kekuatan besar inferensi Bayesian dalam tindakan. Dan kekuatan besar pendekatan Bayes net adalah bahwa probabilitas yang dihasilkan pada setiap tahap pengembangan pengetahuan secara matematis dan ilmiah terdengar. Dengan kata lain, mengingat pengetahuan apa pun yang kita miliki tentang pasien kita, maka berdasarkan pengetahuan matematis dan statistik terbaik sampai saat ini, network akan memberi tahu kita apa yang dapat kita simpulkan secara sah. Ini adalah alat yang sangat ampuh. Luangkan waktu untuk memikirkannya. Anda sebagai dokter tidak hanya mengandalkan firasat, atau rasa intuitif tentang kemungkinan penyakit, seperti yang mungkin Anda alami di masa lalu, namun, berdasarkan perkiraan ilmiah tentang kemungkinan penyakit secara ilmiah dan terbukti akurat, penyakit yang mendapat lebih banyak dan lebih akurat saat Anda mendapatkan pengetahuan tentang pasien tertentu, atau tentang populasi tertentu yang berasal dari pasien tersebut.

Jadi, mari kita lihat bagaimana menambahkan pengetahuan tentang pasien tertentu yang menyesuaikan probabilitasnya. Mari kita katakan seorang wanita masuk, pasien baru, dan kita mulai berbicara dengannya. Dia mengatakan bahwa dia sering kekurangan nafas (dyspnea). Jadi, kita memasukkan temuan itu ke dalam network kita. Dengan Netica kita akan melihat, ini sederhana mengarahkan mouse ke node dan mengkliknya sekali, dimana sebuah daftar keadaan yang tersedia muncul, dan Anda kemudian klik pada item yang benar dalam daftar. Setelah melakukan itu, inilah network itu. Perhatikan bagaimana kotak Dyspnea berwarna abu-abu, menunjukkan bahwa kita memiliki bukti bahwa benda itu ada di salah satu statenya. Dalam cases ini, karena pasien kita tampak dapat dipercaya, kita katakan bahwa kita 100% yakin bahwa pasien kita menderita dispnea. Sangat mudah dengan Netica untuk memasuki temuan yang tidak pasti (juga disebut temuan kemungkinan/ **likelihood finding**), katakanlah 90% Present, tapi mari kita tetap pada hal-hal yang sederhana untuk saat ini.



Amati bagaimana dengan temuan baru ini, bahwa pasien kita mengalami dyspnea, bahwa probabilitas untuk ketiga penyakit tersebut telah meningkat. Kenapa ini? Nah, karena semua penyakit tersebut memiliki dispnea sebagai gejala, karena pasien kita memang menunjukkan gejala ini, hanya masuk akal bahwa keyakinan kita akan kemungkinan adanya penyakit tersebut harus meningkat. Pada dasarnya, adanya gejala tersebut telah meningkatkan kepercayaan kita bahwa dia mungkin sakit parah.

Mari kita lihat inferensi/ kesimpulan mereka lebih dekat.

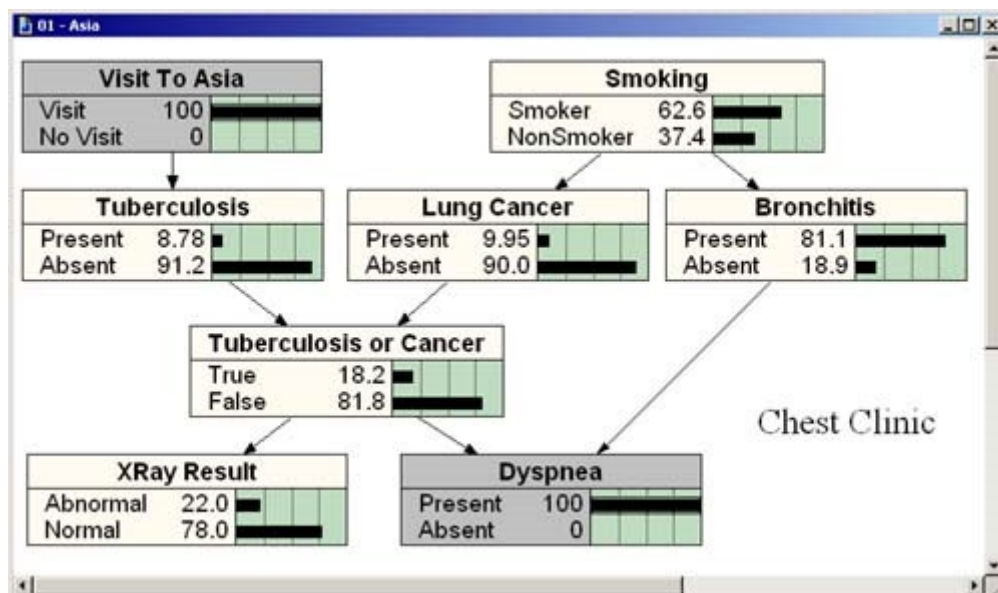
1. Lompatan yang paling signifikan adalah Bronchitis, dari 45% menjadi 83,4%. Mengapa lompatan besar seperti itu? Nah, bronchitis jauh lebih umum daripada kanker atau TB. Jadi, begitu kita memiliki bukti penyakit paru-paru yang serius, ini menjadi diagnosis kandidat kita yang paling mungkin.
2. Kemungkinan bahwa pasien kita adalah perokok kini telah meningkat secara substansial, dari 50% sampai 63,4%.

3. Kemungkinan bahwa dia baru saja mengunjungi Asia telah meningkat sangat sedikit: dari 1% menjadi 1,03%, yang tidak signifikan.
4. Kemungkinan mendapatkan X-Ray abnormal dari pasien kita juga naik sedikit, dari 11% menjadi 16%.

Jika Anda memikirkan perluasan pengetahuan kita, ini sangat membantu. Kita hanya memasukkan satu temuan, adanya Dyspnea, dan pengetahuan ini telah "menyebarkan/ propagated" atau menyebarkan pengaruhnya ke seluruh network, memperbarui semua kemungkinan beliefs/ keyakinan lainnya secara akurat. Beberapa kepercayaan kita meningkat secara substansial, yang lain hampir tidak ada sama sekali. Dan keindahannya adalah jumlah itu justru dihitung.

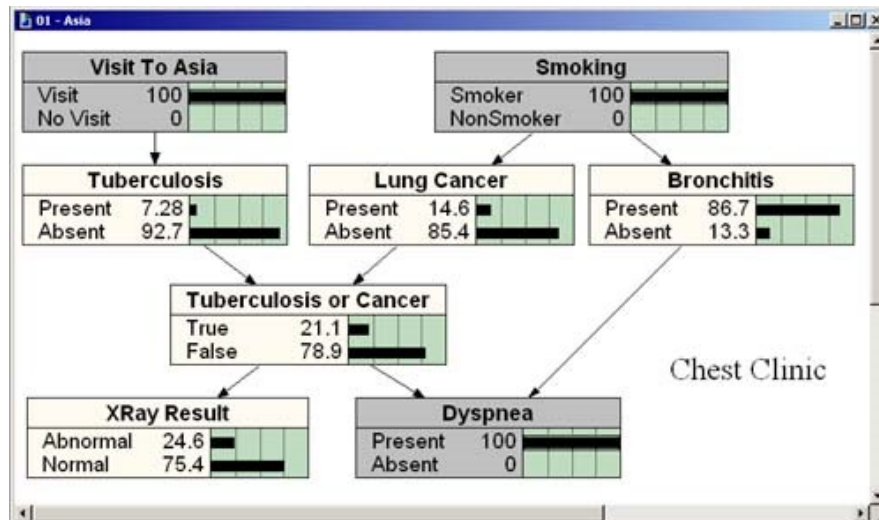
Kita masih belum tahu apa tepatnya sakitnya pasien kita. Keyakinan terbaik kita saat ini adalah bahwa dia menderita Bronchitis (probabilitas Hadir/ Present = 83,4%). Namun, kita ingin meningkatkan kemungkinan diagnosis yang benar. Jika kita berhenti disini dan mendiagnosisnya dengan Bronchitis dan dia benar-benar menderita Kanker, pastilah kita adalah dokter yang malang. Kita benar-benar membutuhkan lebih banyak informasi.

Jadi, dengan teliti, kita menelusuri daftar pertanyaan standar kita. Kita bertanya apakah dia pernah ke Asia baru-baru ini. Anehnya, dia menjawab "ya". Sekarang, mari kita lihat bagaimana pengetahuan ini mempengaruhi network.

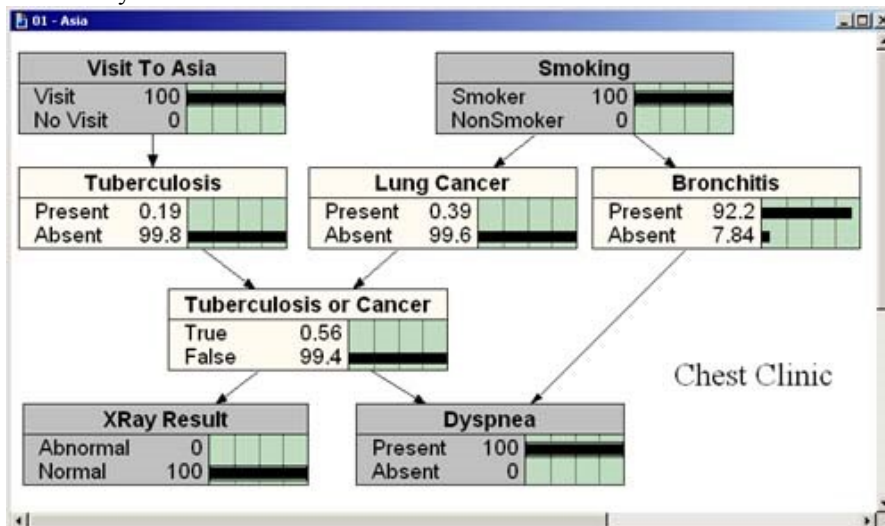


Tiba-tiba, kemungkinan tuberkulosis meningkat secara signifikan, dari 2% sampai 9%. Perhatikan, yang menarik, kemungkinan kanker paru-paru, bronchitis, atau pasien kita menjadi perokok semuanya mengalami penurunan. Kenapa ini? Nah, ini karena penjelasan tentang dyspnea sekarang lebih kuat dijelaskan oleh tuberkulosis daripada sebelumnya (walaupun bronchitis masih tetap menjadi diagnosis kandidat terbaik). Dan karena kanker dan bronchitis sekarang kurang mungkin, begitu juga merokok. Fenomena ini disebut "menjelaskan jauh/ *explaining away*" dilingkaran Bayes net. Dikatakan bahwa ketika Anda memiliki kemungkinan penyebab yang bersaing untuk beberapa peristiwa, dan kemungkinan salah satu penyebabnya meningkat, kemungkinan penyebab lainnya harus menurun karena mereka "*explaining away*" oleh penjelasan pertama.

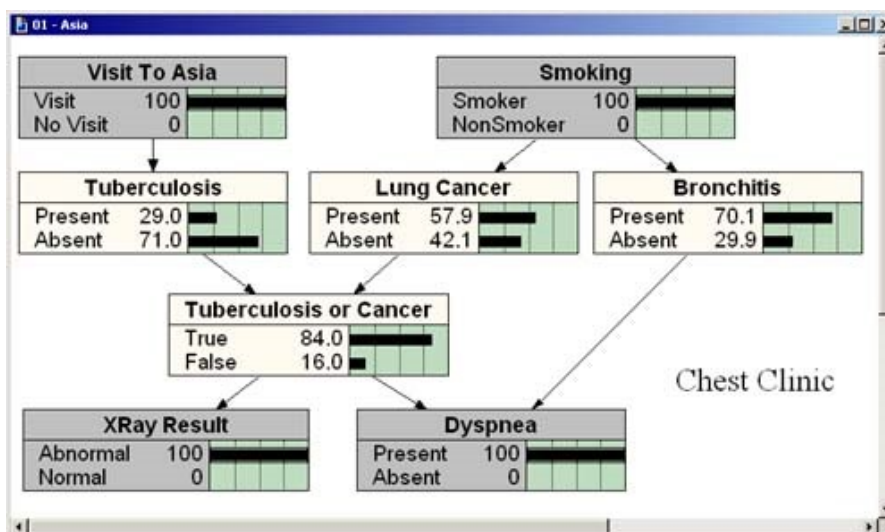
Untuk melanjutkan contoh kita, anggaplah kita mengajukan lebih banyak pertanyaan dan mengetahui bahwa pasien kita memang seorang perokok. Berikut adalah net yang diperbarui.



Perhatikan bahwa hipotesis terbaik kita saat ini masih tetap bahwa pasien menderita Bronchitis, dan bukan kanker TB atau paru-paru. Tapi yang pasti, kita melakukan X-Ray diagnostik. Mari kita katakan bahwa sinar-X ternyata normal. Hasilnya adalah:



Perhatikan bagaimana ini lebih kuat mengkonfirmasi Bronchitis dan disconfirms TB atau kanker paru-paru. Tapi misalkan sinar X tidak normal/ abnormal. Hasilnya adalah:



Perhatikan perbedaan besar. Kanker TB atau Paru telah sangat melonjak probabilitasnya. Bronkitis masih merupakan penyakit yang paling mungkin terjadi pada tiga penyakit yang berbeda, namun kurang dari kombinasi hipotesis TB atau Kanker Paru. Jadi, kita kemudian akan memutuskan untuk melakukan tes lebih lanjut, memesan tes darah, biopsi network paru, dan sebagainya. Bayes nets kita saat ini tidak mencakup pengujian tersebut, namun akan mudah untuk memperpanjangnya hanya dengan menambahkan nodes ekstra saat kita memperoleh statistik baru untuk prosedur diagnostik tersebut. Dan kita tidak perlu membuang bagian network sebelumnya. Ini adalah fitur kuat lainnya dari Bayes nets. Mereka mudah diperluas (atau dikurangi, disederhanakan) agar sesuai dengan kebutuhan Anda yang berubah dan perubahan pengetahuan Anda.

Ringkasan

Pada bagian ini, kita mengetahui bahwa network Bayesian adalah cara matematis yang ketat untuk menjadi model dunia, yang fleksibel dan mudah beradaptasi dengan tingkat pengetahuan apa pun yang Anda miliki, dan yang efisien secara komputasi.

1.1.2 Membantu Pengambilan Keputusan

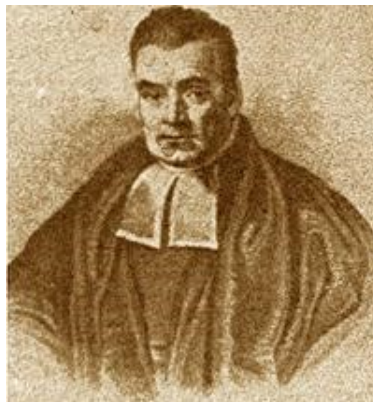
Ini adalah satu hal untuk memprediksi kenyataan seakurat mungkin, namun perluasan alami dan sangat berguna dari hal ini hanyalah untuk menimbang keadaan model Anda dengan tingkat "kebaikan" atau "keburukan" ("goodness" or "badness"). Artinya, jika beberapa state di dunia mengarah pada kesenangan, sementara yang lain merasa sakit, Anda hanya ingin mengetahui bagaimana Anda bisa mengubah dunia untuk memaksimalkan kesenangan dan meminimalkan rasa sakit. Tentu saja, Anda bisa menggunakan istilah lain untuk nilai, selain kesenangan dan rasa sakit, seperti uang, waktu luang, peningkatan kelangsungan hidup, dan sebagainya. Ada ilmu pengambilan keputusan yang mencampurkan probabilitas dengan pengukuran nilai. Itu disebut **Decision Theory** or **Utility Theory**. Bayes nets mudah diperluas ke utilitas komputasi, mengingat tingkat pengetahuan yang kita miliki pada situasi, dan karena itu mereka menjadi sangat populer dalam pembuatan keputusan bisnis dan civic/ kepentingan umum seperti scientific dan economic modeling. Kita akan melihat beberapa contoh ini nanti ditutorial saat kita menggunakan Netica untuk pengambilan keputusan.

Catatan tentang terminologi. Bayes nets yang digunakan secara ketat untuk pemodelan kenyataan/ reality sering disebut "**belief nets**", sedangkan yang juga mencampur unsur value dan decision making, seperti "**decision nets**". Tentu saja, Anda dapat menggunakan belief untuk membuat keputusan, namun dalam keputusan yang benar, keputusan yang benar diantara opsi yang diberikan dihitung untuk Anda, dengan alasan kuantitatif. Network itu sendiri memberi Anda keputusan yang optimal. Jika Anda memilih untuk bertindak berbeda dari yang disarankan, itu pasti karena Anda memiliki informasi tambahan yang tidak terwakili didalam net, atau Anda merasa tidak memutuskan secara rasional. Ini tentu saja mengasumsikan bahwa orang yang rasional akan ingin memaksimalkan, kesenangan, uang, atau ukuran nilai apa pun yang Anda pilih, yang merupakan pertanyaan yang diperdebatkan diantara para filsuf. Tapi kita tidak akan masuk arena itu disini.

Salah satu aplikasi menarik dari *decision net* adalah pada pengendali robot. Robot tidak hanya menghitung tindakan terbaik menggunakan Bayes nets, tapi sebenarnya melakukan tindakan itu.

1.2. Mengapa Bayes Nets disebut Bayes Nets?

Bayes nets adalah network hubungan (networks of relationships), karenanya nets. Dan mereka diberi nama "Bayes" setelah Pendeta Thomas Bayes, 1702-1761, seorang teolog Inggris dan matematikawan yang menuliskan sebuah undang-undang dasar probabilitas yang sekarang disebut Bayes rule.



Bayes Rule :

Untuk dua peristiwa/ events, A dan B,
 $p(B | A) = p(A | B) \times p(B) / p(A)$

dimana Anda membaca 'p(A)' sebagai "probabilitas A", dan
'p(A | B)' sebagai "probabilitas A mengingat bahwa B telah terjadi".

Ternyata Bayes rule sangat kuat dan merupakan aturan perhitungan dasar yang memungkinkan kita untuk memperbarui semua probabilitas dalam net, ketika ada satu informasi yang berubah. Inilah contohnya.

Misalkan Anda tinggal di London, Inggris, dan Anda memperhatikan bahwa selama musim dingin, hujan turun/ *rains* 50% dari waktu dan mendung/ *cloudy* 80% dari waktu (terkadang mendung tanpa hujan). Anda tentu saja tahu bahwa 100% dari waktu, jika sedang hujan, maka itu juga mendung. Menurut Anda, apa kemungkinan hujan, karena itu hanya mendung? Nah, aturan Bayes memungkinkan Anda untuk menghitung ini.

Aturan Bayes mengatakan itu

$$p(R | C) = p(R)p(C | R) / p(C) = 0,5 \times 1,0 / 0,8 = 0,625 = 5/8.$$

Jadi, 5/8 dari waktu, di London selama musim dingin, jika mendung, maka itu hujan.

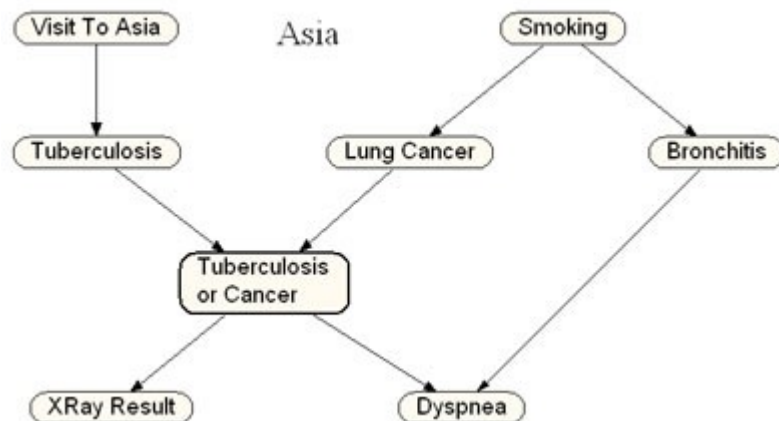
Aturan Bayes dapat diperluas ke beberapa variabel dengan banyak status. Persamaan itu jauh lebih kompleks untuk ditulis, dan sulit dihitung dengan tangan, tapi mudah untuk dipecahkan komputer, itulah salah satu alasan mengapa program seperti Netica sangat berharga.

1.3 Bayes Nets digunakan untuk apa?

Bayes nets dapat digunakan dalam setiap jalan kehidupan dimana pemodelan realitas yang tidak pasti dilibatkan (dan karenanya probabilitasnya ada), dan, dalam cases decision nets, dimanapun sangat membantu untuk membuat keputusan yang dapat dipercaya, dapat dibenarkan, dapat diukur yang akan memaksimalkan kemungkinan/ kesempatan/ chances dari hasil yang diinginkan. Singkatnya, Bayes nets berguna dimana-mana.

1.3.1 Diagnosis

Asia net adalah diagnostic net yang khas.



Dua node atas adalah untuk predispositions/ kecenderungan yang mempengaruhi kemungkinan penyakit. Penyakit tersebut muncul dibawahnya. Dibagian bawah ada gejala penyakit. Untuk tingkat yang besar, hubungan network sesuai dengan sebab-akibat. Ini adalah struktur umum untuk network diagnostik: nodes predisposisi dibagian atas, dengan link ke nodes yang mewakili kondisi internal dan keadaan kegagalan, yang pada gilirannya memiliki link ke nodes untuk pengamatan/ observables. Seringkali ada banyak lapisan node yang mewakili kondisi internal, dengan hubungan antara keduanya mewakili complex inter-relationships nya.

Diagnosisnya bisa bersifat medis atau mekanis. Banyak aplikasi industri Bayes nets adalah untuk menentukan kemungkinan kegagalan komponen. Industri nuklir, industri penerbangan, industri konstruksi, dimana pun kehidupan dan uang dipertaruhkan, semuanya adalah domain alami untuk menerapkan Bayes nets.

1.3.2 Prediksi

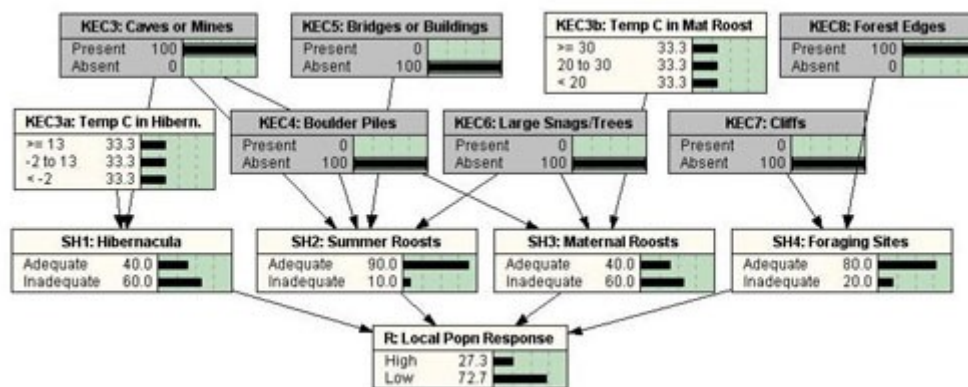
Karena Bayes nets secara alami merupakan rantai sebab-akibat/ *causal chains*, yaitu hubungan dapat menjadi hubungan sebab-akibat antara node induk dan anak, Anda dapat memberi bukti/ evidence kejadian/ events masa lalu, dan kemudian menjalankan Bayes nets untuk melihat hasil kemungkinan masa depan yang akan terjadi. Bayes nets digunakan untuk peramalan cuaca, prediksi pasar saham, pemodelan ekologi, dan lain-lain, untuk membuat prediksi semacam itu. Kekuatan mereka adalah mereka sangat kuat terhadap informasi yang hilang, dan akan membuat perkiraan terbaik dengan informasi apa pun yang ada.

1.3.3 Manajemen Risiko Keuangan, Alokasi Portofolio, Asuransi

Bank officers, penjamin asuransi, investment advisors, semua perlu membuat keputusan yang sulit, dimana seringkali semua faktor yang mempengaruhi cases tidak diketahui. Dengan Bayes nets, mereka masih dapat membuat keputusan yang cerdas, dapat diukur, dan dapat dibenarkan, dengan informasi apapun yang tersedia.

1.3.4 Pemodelan Ekosistem

Bayes nets banyak digunakan dalam pemodelan ekosistem. Seringkali ahli ikan dan satwa liar dihadapkan pada tugas yang sulit untuk menyarankan kebijakan penggunaan lahan. Mereka harus menyeimbangkan kepentingan industri, komunitas, dan alam dan mereka membutuhkan argumen yang masuk akal dan dapat dibenarkan secara ilmiah untuk mendukung analisis dan keputusan mereka. Dengan Bayes nets mereka dapat memodelkan ekosistem dan mendapatkan probabilitas suara apakah spesies tertentu terancam oleh perkembangan industri tertentu. Misalnya, dibawah ini adalah Bayes nets yang menggambarkan bagaimana populasi kelelawar bertelinga Townsend dikaitkan dengan berbagai habitat yang tersedia, suhu mereka selama waktu berkembang biak atau waktu mode hibernasi, dan sebagainya. Ini diambil dari Bruce G. Marcot, dari Dinas Kehutanan AS, dalam makalahnya: Menggunakan Bayesian Belief Networks untuk Mengevaluasi Viabilitas Penduduk Ikan dan Satwa Liar di bawah Alternatif Pengelolaan Lahan dari Pernyataan Dampak Lingkungan.



1.3.5 Sensor Fusion

Sensor fusion mengacu pada kelas masalah dimana data dari berbagai sumber harus diintegrasikan untuk sampai pada interpretasi suatu situasi. Misalnya, data dari berbagai kamera yang diambil dari berbagai sudut dan resolusi mungkin perlu diintegrasikan untuk menentukan apa yang ada dalam sebuah adegan. Atau sensor industri masing-masing dapat melaporkan keadaan sebuah mesin, dan hanya dengan menggabungkan semua pembacaannya, mereka bisa mendapatkan gambaran yang lengkap. Seringkali dalam masalah sensor fusi, seseorang harus menghadapi resolusi temporal atau spasial yang berbeda, dan seseorang harus memecahkan "masalah korespondensi", yaitu menentukan kejadian dari satu sensor yang sesuai dengan kejadian yang sama seperti yang dilaporkan pada sensor lainnya. Karena Bayes nets kuat terhadap data yang hilang, mereka menggabungkan informasi dengan baik. Jadi, sementara masing-masing sensor hanya memiliki kesempatan terbatas untuk memberikan interpretasi yang benar, kombinasi dari semua peluang sensor, biasanya meningkatkan kemungkinan interpretasi yang valid. Di bidang *computer and robotic vision*, Bayes nets telah banyak digunakan. Demikian juga untuk pemahaman citra sonar.

1.3.6 Monitoring dan Alerting/ Peringatan

Perluasan untuk mendiagnosis sebuah sistem, memutuskan kapan harus mengirim peringatan bahwa sistem berada dalam keadaan tidak sehat. Seringkali ada biaya yang parah jika sistem gagal dan juga biaya yang mahal untuk mengirimkan alarm palsu (pancaran diacak, jalur perakitan dimatikan, dll.). Bayes nets, sebagai decision nets, adalah hal yang wajar untuk membuat keputusan terbaik dengan data sensor yang ada. *Vista system* dari Eric Horvitz, "adalah decision nets yang digunakan di Pusat Kontrol Misi NASA di Houston atau beberapa tahun. Sistem ini menggunakan network Bayesian untuk menafsirkan telemetri langsung dan memberikan saran mengenai kemungkinan kegagalan sistem penggerak ruang angkasa lainnya." (ref # 1)

Netica API Toolkit menawarkan semua tools yang diperlukan untuk membangun aplikasi semacam itu.

Ringkasan

Bayes nets memiliki potensi untuk diaplikasikan cukup banyak dimana-mana.

1.4 Sifat Menarik/ Interesting Properties dari Bayes Nets

1.4.1 Probabilitas tidak perlu tepat untuk menjadi berguna


Beberapa orang telah menghindari penggunaan Bayes nets karena mereka membayangkan mereka hanya akan bekerja dengan baik, jika probabilitas yang mendasari mereka tepat. Ini tidak benar. Ternyata sangat sering bahwa perkiraan probabilitas, bahkan yang subjektif yang bisa ditebak, memberikan hasil yang sangat bagus. Bayes nets umumnya cukup kuat untuk pengetahuan yang tidak sempurna. Seringkali kombinasi beberapa helai pengetahuan tidak sempurna memungkinkan kita membuat inferensi/ kesimpulan kuat yang mengejutkan.

1.4.2 Probabilitas Kondisional Kausal lebih mudah diperkirakan daripada sebaliknya

Studi telah menunjukkan orang lebih baik dalam memperkirakan probabilitas "ke arah depan". Misalnya, dokter cukup pandai dalam memberikan perkiraan probabilitas untuk "jika pasien menderita kanker paru-paru, berapakah kemungkinan sinar-X mereka abnormal?", Bukan sebaliknya, "jika sinar-X abnormal, Apakah kemungkinan kanker paru-paru penyebabnya?" (Jensen96)

2. Menggunakan Bayes Net

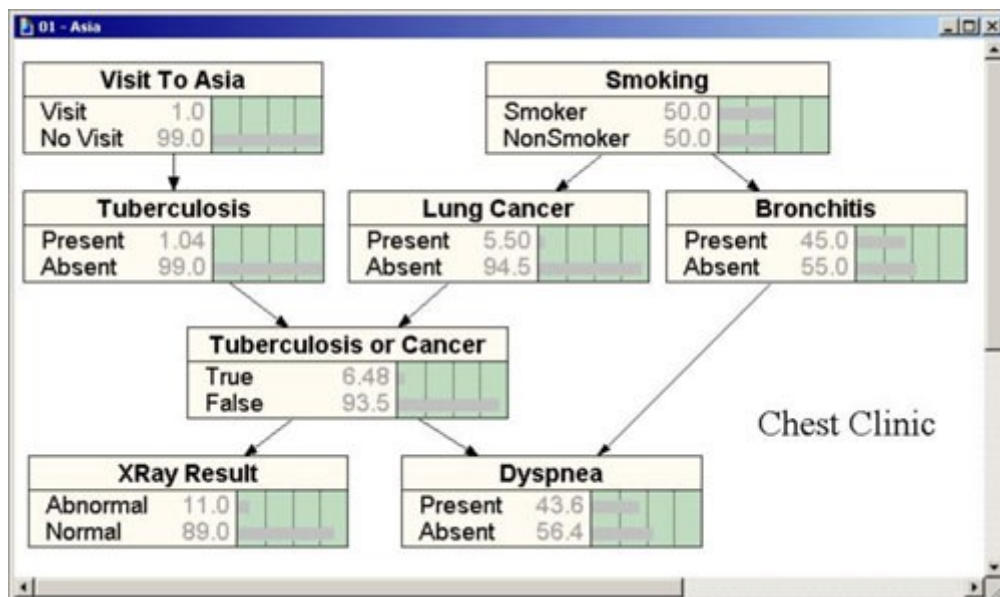
Dalam tutorial ini kita akan melangkah selangkah demi selangkah melalui beberapa operasi yang lebih umum yang biasanya dilakukan pengguna pada Bayes nets. Di Bab 2 kita akan belajar bagaimana membangun network/ net. Untuk saat ini, kita membatasi diri kita untuk menggunakan salah satu yang diberikan kepada kita.


Anda dapat membuka network dengan ekstensi ".dne" hanya dengan mengkliknya didalam file manager Anda. Atau Anda bisa membuka Netica dan membuka file net yang sudah ada sebelumnya menggunakan perintah menu: File-> Open ..., atau tombol tool-bar: 

Silahkan buka Asia.dne.

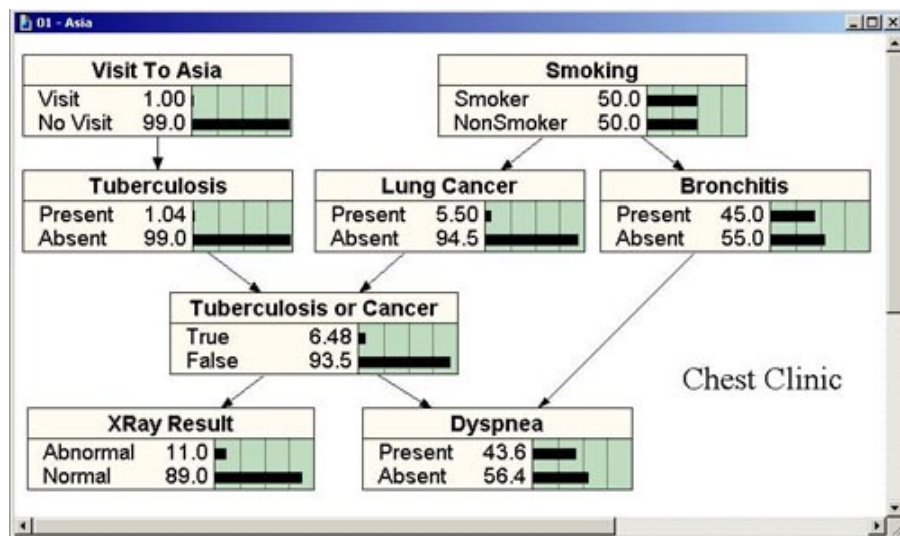
2.1 Kompilasi net

Saat baru dibuka, Bayes nets adalah *uncompiled*. Ini berarti bahwa pohon persimpangan/ *junction tree* untuk inferensi cepat belum dibangun secara internal untuk network. Inilah bagaimana Asia.dne terlihat saat baru dibuka. Perhatikan bahwa hanya probabilitas sebelumnya (probabilitas sebelum bukti dimasukkan) ditampilkan untuk setiap node, dan bahwa belief-bar berwarna abu-abu.



Untuk mendapatkan inferensi/ kesimpulan apa pun, Anda harus mengkompilasi net terlebih dahulu. Hal ini dilakukan dengan perintah menu: Network → compile, atau tool-bar button: 

Setelah kompilasi, belief-bars berubah menjadi hitam, namun sebaliknya, tidak ada perubahan. Ini karena kita belum memasukkan bukti apapun di net. Kita tidak memiliki pengetahuan khusus tentang variabel mana pun di dunia kita. Semua yang kita miliki adalah probabilitas sebelumnya untuk variabel-variabel tersebut untuk digunakan, dan begitulah semua yang dapat ditunjukkan.



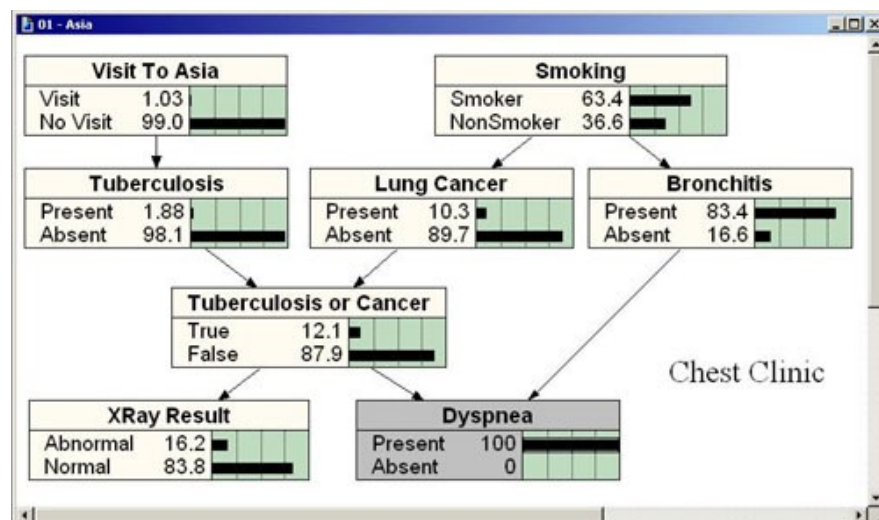
2.2 Memasukan Temuan/ Findings

Temuan dimasukkan dengan mengklik kanan tombol mouse pada node dimana Anda memiliki beberapa informasi temuan. Dialog akan muncul dekat mouse yang memungkinkan Anda untuk mengklik (baik kiri atau kanan tombol mouse) pada tipe informasi temuan/ finding yang dapat dimasukkan untuk node tersebut. Anda juga dapat menggunakan dialog ini untuk menghapus bukti/ evidence dari suatu nilai node: cukup klik "Unknown".



| |
|------------|
| Likelihood |
| ✓ Unknown |
| Present |
| Absent |

2.2.1 Temuan Keadaan Positif/ Positive State Findings

Untuk node yang memiliki sejumlah state secara diskrit (lihat B.1.1 *Discrete vs Continuous Variables*), keadaan yang tersedia ditunjukkan di bawah bilah horizontal. Mengklik pada state tertentu berarti bukti/ evidence kita adalah bahwa dunia kita berada dalam keadaan ini. Ini adalah cara yang paling umum untuk memasukkan temuan. Cobalah memasukkan temuan di Asia.dne dan perhatikan apa yang terjadi pada probabilitas didalam net. Di sini kita menunjukkan Asia setelah kita berbicara dengan pasien kita dan mengetahui bahwa ia mengalami sesak napas (Dyspnea):



Perhatikan bahwa setelah memasukan temuan *Dyspnea = Present/ Hadir*, probabilitas *Dyspnea* untuk pasien kita sekarang 100%. Probabilitas semua node yang tidak terlepas dari *Dyspnea* juga akan diperbarui (pengecualian untuk ini adalah jika Anda telah mengaktifkan "pembaruan otomatis/ auto-updating" untuk network Anda (lihat Auto-Updating). Perhatikan bagaimana sesak napas/ *dyspnea* meningkatkan kemungkinan ketiga penyakit tersebut. Netica mengatur warna latar belakang node dengan bukti/ evidence yang masuk ke abu-abu.

Trik: Untuk mendapatkan tampilan visual yang bagus dari pengaruh suatu temuan/ finding yang ada didalam network, Anda dapat secara bergantian menekan Undo ( atau Ctrl-Z) dan Redo ( atau Ctrl-Shift-Z).

Catatan Terminologi. Setelah beberapa bukti dimasukkan ke dalam network, semua probabilitas dipanggil/ disebut sebuah "*posteriori*" probabilities, yang berarti "setelah" bukti/ evidence yang diberikan. Ini berbeda dengan sebuah *priori* probabilities, yaitu "sebelum" bukti/ evidence.

2.2.2 Temuan Negatif/ Negative Findings

Terkadang Anda tidak tahu secara eksplisit bahwa dunia berada dalam keadaan tertentu, namun Anda tahu bahwa itu tidak dapat terjadi di beberapa state. Netica memungkinkan Anda untuk bekerja dengan tipe pengetahuan negatif semacam ini juga. Misalnya, katakanlah nodes Suhu/ Temperature bisa mengambil nilai dingin, sedang/ medium, dan panas. Anda bisa mendapatkan informasi bahwa suhu tidak panas. Itu adalah temuan negatif tunggal/ single negative finding. Jika Anda menerima temuan negatif lain bahwa suhu not medium, maka Anda bisa menyimpulkan bahwa itu dingin. Jadi beberapa temuan negatif bisa setara dengan satu temuan positif.

Anda bisa memasukkan negative finding information hanya jika network disusun dan ditampilkan dengan gaya belief-bar (menu: Style → Belief-Bars). Untuk melakukan ini, tahan tombol SHIFT saat Anda mengklik nama finding yang Anda tahu adalah tidak. Jika Anda mengklik nama tambahan, mereka juga ditambahkan ke daftar nilai state yang tidak masuk ke node ini. Untuk mengatur ulang informasi negatif, cukup klik kanan dan klik "Unknown".

Biasanya Anda hanya akan repot-repot dengan temuan negatif ketika sebuah node memiliki lebih dari dua state, walaupun legal untuk memasukkan nilai positif dari 2 state (nilai biner) node dengan memasukkan nilai alternatifnya secara negatif.


2.2.3 Kemungkinan Temuan/ Likelihood Findings

Informasi temuan positif disebut "*hard evidence*". Anda tahu pasti keadaan sebuah node. Bukti lain disebut "*soft evidence*". Di situlah Anda tidak tahu nilai node dengan pasti, namun Anda memiliki gagasan tentang nilai apa yang dapat diambilnya. Temuan negatif adalah bentuk *soft evidence*. Bentuk lain yang lebih umum, adalah dimana Anda menentukan "*likelihood distribution*" untuk nilai node. Ini hanyalah sebuah *array* dari probabilitas, satu untuk setiap kemungkinan keadaan node.

Untuk memberi *likelihood finding*, cukup klik kanan pada node, dan klik "*Likelihood*". Sebuah dialog akan muncul yang akan meminta Anda untuk memasukkan nilai probabilitas untuk keadaan pertama. Setelah memasukan itu, Anda akan dimintai probabilitas untuk keadaan kedua, dan seterusnya.

2.3 Menghapus Temuan/ Removing Findings

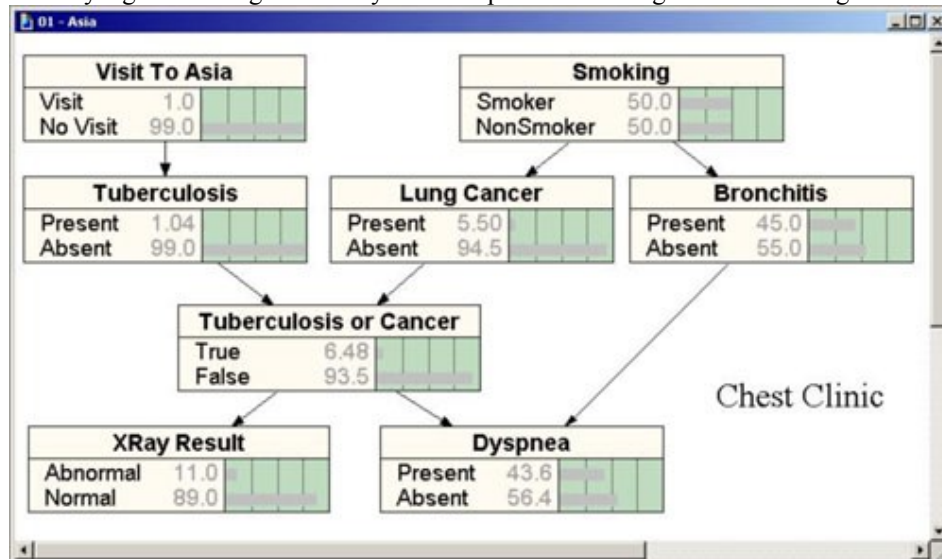
Jika Anda ingin menghapus temuan untuk node tertentu, klik tombol kanan di atasnya dan pilih "Unknown".

Jika Anda ingin menghapus semua temuan di keseluruhan network (mungkin pasien baru saja masuk), pilih Network-> Remove Findings, atau gunakan tombol shortcut: .

3. Inferensi Probabilistik

Tutorial ini memperluas topik yang diperkenalkan pada tutorial sebelumnya. Kita akan membahas beberapa pemikiran dan isu umum yang harus Anda ketahui dalam memeriksa Bayes nets dan menggunakannya untuk menarik inferensi/ kesimpulan.


Semua informasi yang terkandung dalam Bayes nets dapat diamati dengan memeriksa tiga hal.

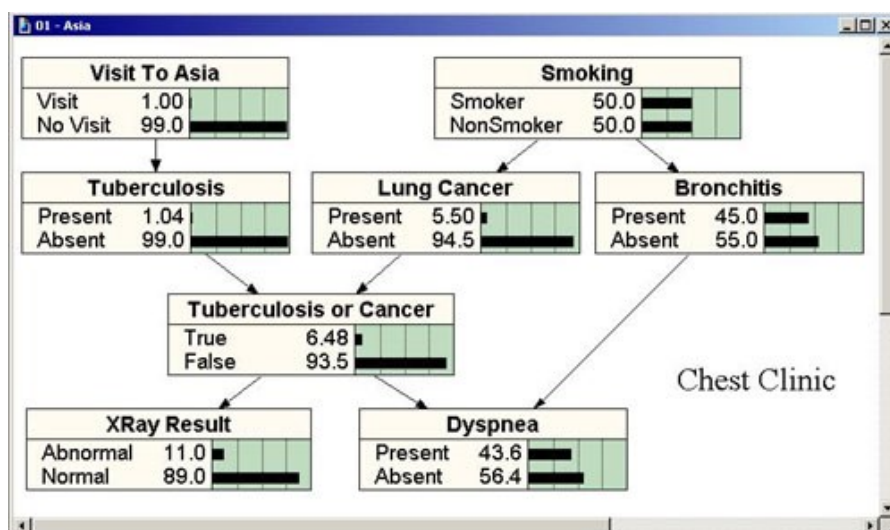


1. Pertama, ada struktur network, yang terdiri dari node dan link mereka, yang bisa Anda lihat dalam diagram network yang saat ini ditampilkan.
2. Kedua, adalah sifat/ properties dari setiap node, yang dapat Anda lihat di kotak dialog node mereka, diperoleh dengan mengklik dua kali pada node.
3. Ketiga, adalah hubungan sebenarnya/ actual relationships diantara node. Inilah probabilitas bersyarat/ conditional probabilities yang menghubungkan dua node.

3.1 Kompilasi, ditinjau ulang/ revisited

Sebelum Anda bisa melakukan inferensi apapun di net, Anda harus **compile**/ mengkompilasi net. Kompilasi network besar bisa memakan banyak waktu, dan jika pembaruan otomatis dihidupkan, setiap kali Anda menambahkan bukti/ evidence, network akan diperbarui sepenuhnya. Oleh karena itu, didalam network besar, atau komputer yang sangat lambat, Anda mungkin ingin menambahkan semua bukti Anda sebelum melakukan langkah kompilasi.

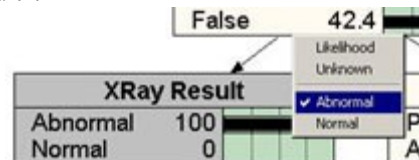
Untuk mengkompilasi network untuk digunakan, klik di jendela untuk membuatnya aktif, dan pilih Network-> Compile (atau gunakan tombol shortcut: ).



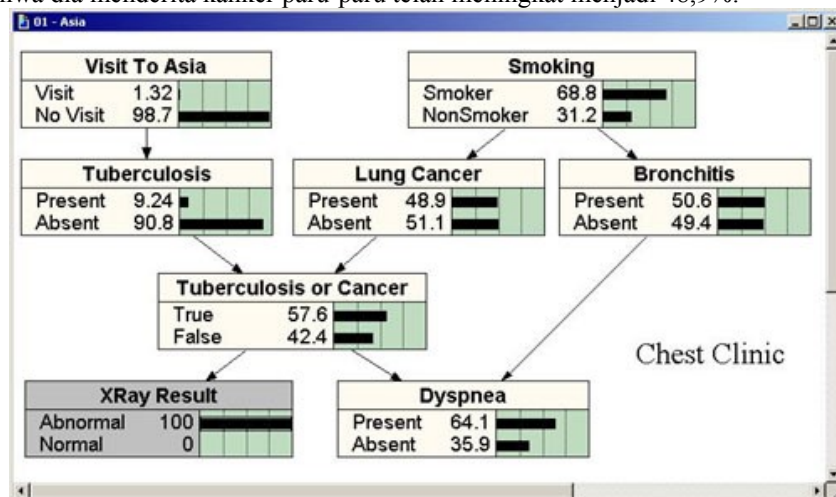
Setelah kompilasi, struktur data yang sesuai untuk inferensi cepat akan dibangun secara internal, dan batang disetiap node akan menjadi gelap, menunjukkan bahwa mereka dan angka di sampingnya sekarang adalah data yang valid. Mereka menunjukkan probabilitas masing-masing keadaan node.

Misalkan kita ingin "mendiagnosis" pasien baru. Saat pertama kali memasuki klinik, tanpa mendapat informasi tentang dia, kita yakin dia menderita kanker paru-paru dengan probabilitas 5,5%, seperti yang dapat dilihat pada Lung Cancer nodes/ Kanker Paru (jumlahnya mungkin lebih tinggi daripada populasi umum, karena sesuatu telah membawanya ke klinik dada/ chest clinic). Ini adalah poin penting. Sebelum Anda dapat menarik inferensi/ kesimpulan yang benar dari penggunaan Bayes nets, pastikan bahwa Anda memahami alasan/ reasoning yang masuk ke konstruksi network, dan dari populasi sampel apa, probabilitas awal atau prior diambil.

Jika pasien kita memiliki x-ray yang abnormal, informasi tersebut dapat dimasukkan dengan mengklik kata "Abnormal" dari nodes "XRay Result":

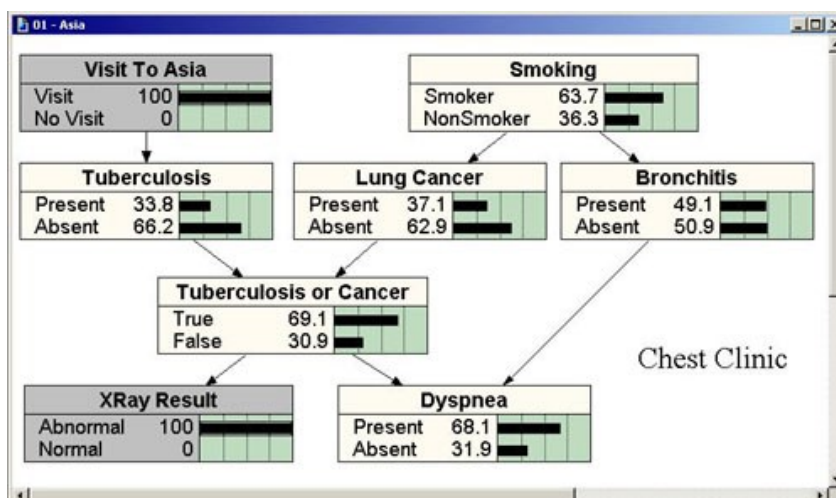


(dalam real-word belief network, Anda mungkin bisa enter dengan cara apa sinar x itu "abnormal"). Semua angka probabilitas dan bar akan berubah untuk memperhitungkan temuan tersebut. Sekarang probabilitas bahwa dia menderita kanker paru-paru telah meningkat menjadi 48,9%:



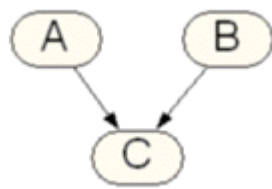
3.2 Menjelaskan/ Explaining Away

Jika selanjutnya Anda menunjukkan bahwa dia telah melakukan kunjungan ke Asia, dengan mengklik "Visit", probabilitas kanker paru-paru turun menjadi 37,1%, karena XRay yang abnormal sekarang sebagian dijelaskan/ *explained away* oleh peluang Tuberkulosis yang lebih besar (yang dapat dia tangkap diAsia):



Catatan: sistem pakar medis kuno memiliki masalah dengan tipe reasoning/ penalaran ini, karena masing-masing temuan "Abnormal XRay" dan "Visit to Asia" sendiri meningkat, atau membiarkan hal yang sama, probabilitas dari kanker paru-paru.

Pola umum untuk "explaining away" adalah sebagai berikut:

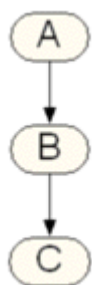


Sebelum kita memiliki evidence untuk C, A dan B adalah independen. Artinya, mengubah satu tidak berpengaruh pada yang lain. Tapi begitu kita memiliki evidence untuk C, setiap perubahan probabilitas A memiliki perubahan yang berlawanan pada B dan sebaliknya. Mereka adalah competing explanations/ penjelasan yang bersaing dari C. Hal ini karena jika kita memperoleh evidence A, maka itu menjelaskan bahwa C, dan peluang B berkurang. Ini adalah fitur penting dari causal reasoning/ penalaran kausal yang baik. Ini mengikuti secara alami dari

hukum statistik Bayesian.

3.3 Pemblokiran Serial/ Serial Blocking

Pemblokiran serial adalah fenomena dimana kepala dari suatu rantai sebab-akibat/ causal chain dapat diblokir dari mempengaruhi ekor, bila bukti pasti diperoleh untuk node intervensi/ campur tangan.

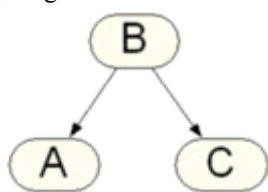


Dengan asumsi bahwa kita tidak yakin/ uncertain tentang B, jika kita mendapatkan bukti untuk C, yang meningkatkan chance/ peluang B, yang pada gilirannya akan meningkatkan peluang A. Namun, jika B ditentukan, yaitu kita mengamati keadaannya (dengan pasti (yaitu probabilitasnya berada dalam keadaan teramati adalah 100%)), maka setiap perubahan pada C tidak dapat mempengaruhi A. Juga tidak ada perubahan pada A yang mempengaruhi C. Jadi, kita mengatakan bahwa kepastian B "block/ menghalangi" ketergantungan apapun sebelumnya dibagi/ shared diantara A dan C. Tepatnya, mungkin ada ketergantungan yang diperkenalkan oleh hubungan lain didalam net, namun tidak melalui B.

Catatan Terminologi. Kita juga dapat menggambarkan situasi pemblokiran serial dengan mengatakan bahwa "C conditionally independent dari A, diberikan B."

3.4 Pemblokiran Divergent/ Divergent Blocking

Pemblokiran yang berbeda/ berlainan/ divergent mirip dengan pemblokiran serial, namun terjadi saat subnet adalah salah satu node yang berbeda:



Dengan asumsi bahwa kita tidak yakin/ uncertain dengan B, jika kita mendapatkan bukti untuk A, yang meningkatkan chance/ peluang B, yang pada gilirannya akan meningkatkan peluang C. Namun, jika B ditentukan, yaitu kita mengamati keadaannya (dengan pasti (yaitu probabilitasnya berada dalam keadaan teramati adalah 100%)), maka setiap perubahan pada A tidak dapat mempengaruhi C. Demikian juga, secara simetri, tidak ada perubahan pada C yang dapat mempengaruhi A. Jadi, kita mengatakan bahwa kepastian B "block" ketergantungan apapun yang sebelumnya dimiliki oleh anak-anaknya. Sekali lagi, tepatnya, mungkin ada ketergantungan yang diperkenalkan oleh hubungan lain di net, tapi tidak melalui B.

Catatan Terminologi. Kita juga dapat menggambarkan situasi pemblokiran divergent dengan mengatakan bahwa "A conditionally independent dari C, diberikan B."

Saat melihat Bayes nets, ada baiknya Anda menyadari potensi untuk "menjelaskan/ explaining away" atau blocking/ memblokir. Ini memungkinkan Anda melihat bagaimana informasi evidential/ jelas dapat mengalir (merambat) melalui net.

4. Dasar Pengambilan Keputusan/ Basic Decision Making

Tutorial ini memperkenalkan *decision nets*. Ini adalah Bayes nets dimana bentuk *utility* (nilai, kebaikan/ value, goodness) telah diperkenalkan, seperangkat keputusan dimungkinkan, dan diinginkan untuk menemukan keputusan yang memaksimalkan ukuran *utility* secara keseluruhan.

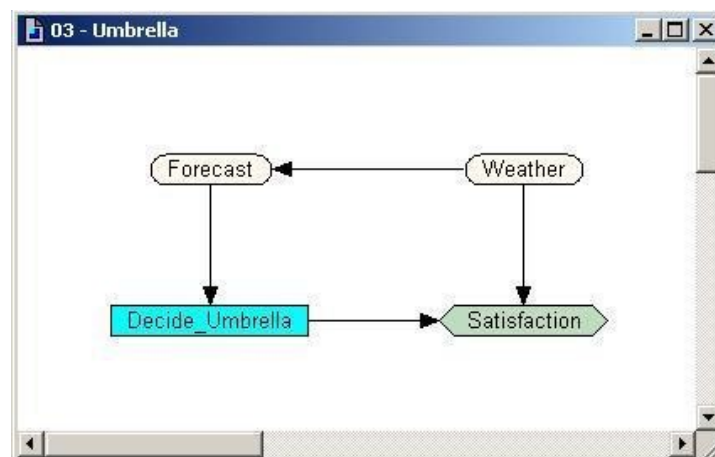
Dengan Netica, *decision nets* sama dengan *Bayes nets* biasa, kecuali dua tipe node lagi hadir,

- *decision nodes*/ node keputusan, dan
- *utility nodes*/ node utilitas .

Dalam Bayes nets biasa, semua node disebut *nature nodes*/ node alami karena hanya berkaitan dengan pemodelan sifat/ nature atau reality/ realitas dunia, kemungkinan/ likelihood keberadaannya berada dalam keadaan yang mungkin terjadi. Konsep *utility* dan konsep *decision* dipandang sebagai diluar/ outside dari hanya menggambarkan realitas, lebih dalam bidang goal/ tujuan, desires/ keinginan, dan agendas/ agenda.

Mari kita lihat *decision net* pertama kita untuk melihat apa *decision* dan *utility* nodes terlihat dan berperilaku seperti.

Network berikut adalah klasik dalam literatur *decision net*. Ini disebut Umbrella, dan karena Ross Shachter. Tujuannya adalah untuk membantu kita memutuskan apakah kita harus mengambil payung kita hari ini, berdasarkan ramalan cuaca hari ini.



Ada link dari Weather/ cuaca ke Forecast/ ramalan yang menangkap korelasi yang meyakinkan diantara keduanya (mungkin berdasarkan pengamatan sebelumnya). Ada link dari Forecast ke Umbrella yang menunjukkan bahwa pembuat keputusan akan mengetahui forecast/ ramalannya saat dia membuat keputusan, tapi tidak ada kaitan dari Weather ke Umbrella, karena jika dia tahu pasti cuaca/ weather apa yang akan terjadi, akan mudah untuk memutuskan apakah akan mengambil payungnya atau tidak.


Network memiliki dua *nature nodes* yang mewakili ramalan cuaca/ weather forecast dipagi hari (cerah, berawan atau hujan/ sunny, cloudy or rainy), dan apakah benar-benar hujan disiang hari (rain or no_rain). Ini memiliki *decision node* apakah perlu mengambil payung, dan *utility node* yang mengukur tingkat kepuasan pembuat keputusan.

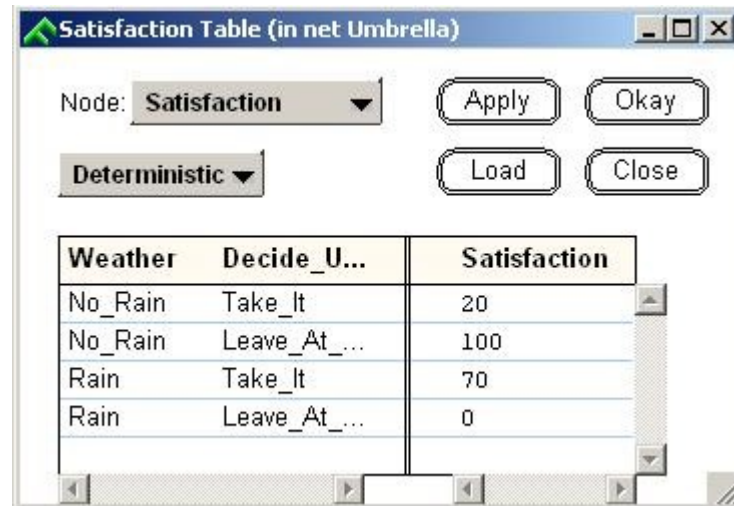
Catatan Visualisasi. Didalam Netica, untuk membantu membedakan mereka, *decision node* dan *utility node* digambar agak berbeda dari *nature node*.

- *utility node* biasanya digambar sebagai bentuk enam sisi, sedangkan
- *decision node* biasanya digambarkan sebagai persegi panjang dengan latar belakang warna yang khas.

Sifat tampilan yang tepat bergantung pada gaya tampilan. Anda dapat mencoba berbagai visualisasi dengan menggunakan menu "Style".

4.1 Utility Nodes

Utility nodes sangat mudah. Mereka berisi tabel dari nilai utilitas (dalam unit apa pun yang Anda inginkan), satu untuk setiap kombinasi induk untuk node itu. Pada dasarnya ini mengatakan, inilah bagaimana kita menilai setiap kemungkinan dunia yang diwakili oleh induk nya. Berikut adalah tabel utilitas untuk node "*Satisfaction*" didalam net kita. (Anda dapat melihat tabel dengan menaikkan dialog node dan mengklik "Table", atau dengan memilih node dan mengklik tombol relasi: )



| Weather | Decide_U... | Satisfaction |
|---------|--------------|--------------|
| No_Rain | Take_It | 20 |
| No_Rain | Leave_At_... | 100 |
| Rain | Take_It | 70 |
| Rain | Leave_At_... | 0 |

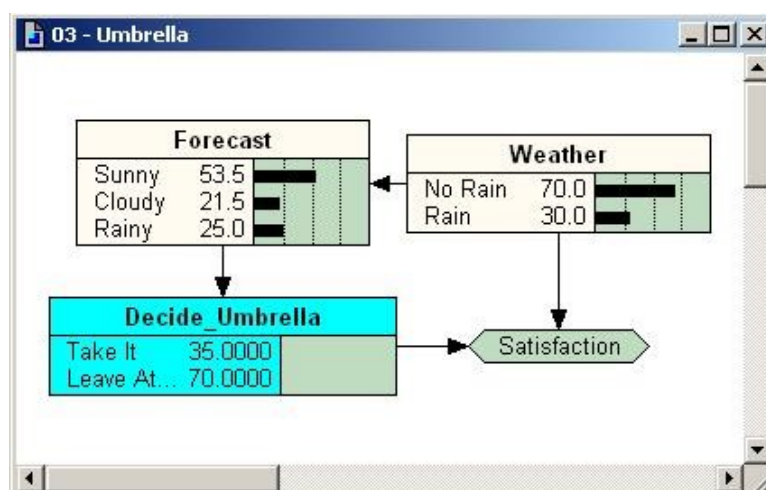
Perhatikan bahwa karena ada kaitan dari Weather and Umbrella to Satisfaction, ada empat entri di tabel kita, satu untuk setiap kombinasi cuaca yang cerah atau hujan (*sunny or rainy*), dibandingkan dengan kita yang menggunakan payung kita atau tidak (*taking our umbrella or not*). Nomor utilitas dalam tabel didasarkan pada skala pribadi kita sendiri tentang relative happiness/ kebahagiaan relatif.

- Kita sangat senang saat tidak hujan dan kita tidak harus mengambil payung, dan kita menetapkan ukuran ini 100.
- Kita paling bahagia saat hujan dan kita mengambil payung: utility = 70.
- Kita tidak suka membawa payung pada hari yang cerah: utility = 20. Dan,
- Kita paling kesal jika sedang hujan dan kita tidak mengambil payung: utility = 0.

Jadi, mari kita kompilasi network dan lihat bagaimana hal itu bisa membantu kita memutuskan untuk mengambil payung atau tidak.

4.2 Decision Nodes

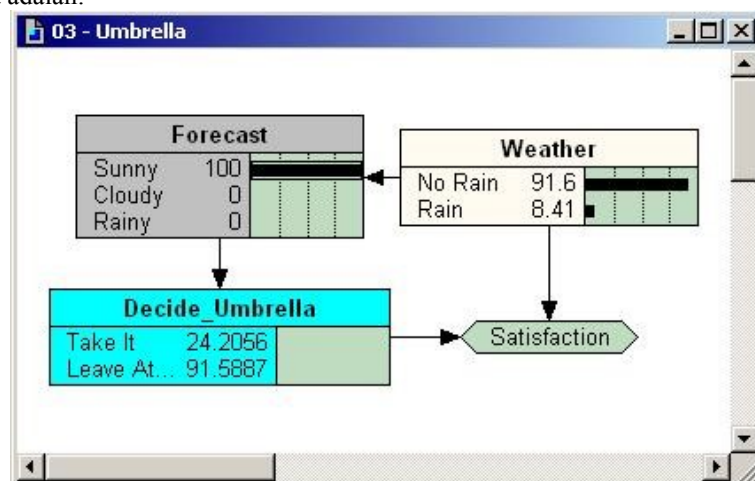
Dengan gaya belief-bar dihidupkan, dan network terkompilasi, network terlihat seperti ini.



Probabilitas yang assigned/ ditugaskan pada weather/ cuaca hanyalah probabilitas awal untuk cuaca di daerah kita. Kita belum menerima perkiraan. Angka disamping setiap pilihan keputusan menunjukkan utilitas yang diharapkan/ expected utility untuk membuat pilihan itu. Konsep "**expected utility**" atau "**expected value**" adalah yang lama. Ini hanyalah jumlah yang Anda dapatkan dengan mengalikan utilitas suatu events/ peristiwa dengan probabilitas kejadian itu. Jadi

$$\text{ExpectedUtility}(A) = \text{Utility}(A) \times p(A).$$

Jadi, dalam melihat decision net kita, kita dapat melihat bahwa, tanpa ramalan cuaca apapun, tanpa bukti/ evidence yang dimasukkan didalam network, nilai yang diharapkan untuk mengambil payung adalah 35, sementara untuk meninggalkannya di rumah adalah 70. Jelas Pilihan terbaik mengingat informasi yang ada adalah meninggalkan payung di rumah. Sekarang anggap kita mendapatkan beberapa informasi. Kita mendengarkan radio dan ramalannya untuk matahari/ sun. Jadi kita masukkan nilai itu di nature node kita, Forecast. Hasilnya adalah:



Coba masukkan nilai untuk setiap tipe ramalan cuaca lainnya dan lihat bagaimana hal itu mempengaruhi keputusan optimal kita, seperti yang digambarkan pada decision node. Decision nets harus memberi tahu Anda bahwa hanya jika ramalannya untuk hujan, sebaiknya kita mengambil payung kita.

Tentu saja, tidak dapat terlalu ditekankan bahwa rekomendasi ini adalah kombinasi dari kedua nature model (natures nodes) dan penilaian yang kita tempatkan pada dunia yang mungkin ada dalam model itu. Jika kita mengubah penilaian kita, keputusan terbaik bisa berubah. Sebagai latihan, coba ubah nilai utilitas pada node Satisfaction. Misalkan Anda adalah pria pegunungan yang mencintai kebebasan, jadi Anda tidak keberatan terlalu basah, tapi Anda benci dibebani oleh payung. Dalam hal ini, taruhan terbaik Anda mungkin tidak akan pernah mengganggu membawa payung, dan lebih baik mengambil kesempatan untuk basah. Itu hanya air. Anda hanya akan kering oleh api!

4.3 Teknik untuk Mendapatkan Penilaian yang Baik/ Good Valuations

Decision nets menggabungkan

- komponen ilmiah (the world, its states, and their relative likelihood/ dunia, keadaannya, dan kemungkinan relatifnya), dan
- komponen nilai (how we like those states/ bagaimana kita menyukai keadaan tersebut).

Mendapatkan pegangan pada komponen ilmiah, meski terkadang sulit, setidaknya merupakan proses yang didefinisikan dengan baik. Ini adalah metode ilmiah yang tepat: definisi yang jelas tentang variabel dan keadaannya, pengukuran terukur dari pengamatan, dll. Mendapatkan pegangan pada komponen nilai terkadang bisa lebih sulit, jika tidak ada cara yang jelas untuk mengukurnya. Terkadang, uang digunakan sebagai ukuran, tapi satu dolar bisa lebih berharga bagi satu orang daripada yang lain, dan nilainya juga relatif terhadap jumlah uang yang sudah dimiliki penerima. Dalam analisis akhir, kita harus mengakui bahwa utilitas dan nilai bersifat subjektif, konsep psikologis dan karenanya secara intrinsik kurang mudah diamati dan diukur. Tapi decision nets hanya akan sama berharganya dengan kualitas komponen ilmiah dan valuasinya, jadi pastikan Anda dapat membenarkan kedua komponen tersebut, sebelum mempercayai net's decision.

Teori keputusan telah menemukan beberapa "trik" untuk membantu orang mengukur nilai subjektif. Berikut adalah beberapa hal sederhana yang mungkin berguna bagi Anda: **Bidding auction**/ Lelang penawaran. Untuk membaca tambahan tentang masalah ini, periksa buku teks yang bagus tentang Teori Keputusan.

B. Operasi Netica Dasar

1. Bagian dari Netica GUI

- Main Window
- Menu Bar
- Tool Bar
- Net Window
- Message Window
- Node Dialog Box

2. Membangun Struktur Net

- Membuat Node
 - Nama dan Judul/ Title
 - Tipe/ Kind
 - Diskrit vs Kontinyu/ Discrete vs. Continuous
 - State
 - Deskripsi dan Properti Lainnya
 - Pembuatan Node Cepat
 - Pembuatan Node Secara Otomatis Dari Case files/ Case Files
- Menambahkan Link
- Memindahkan Node dan AutoGrid
- Memilih Node dan Links
- Membatalkan & Mengulangi/ Undoing & Redoing
- Membentuk Kembali Link/ Reshaping a Link
- Menyimpan dan Memulihkan Network/ Saving and Restoring a Network
- Menghapus Node dan Link
- Penamaan Link
- Mengubah Ukuran Net dan Memperbesar/ Net Resizing and Zooming

3. Mendefinisikan Hubungan Node/ Node Relationships

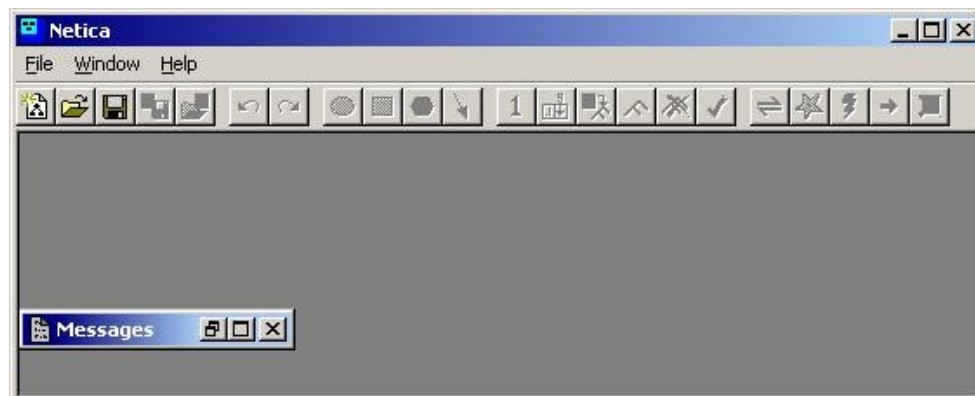
- Mendefinisikan Tabel Probabilitas Secara Manual
- Mendefinisikan Tabel Probabilitas Dengan Persamaan
- Pembelajaran Tabel Probabilitas/ Learning Probability Tables

1. Bagian dari Netica GUI

Dalam tutorial ini kita memperkenalkan Anda ke bagian utama Netica GUI (Graphical User Interface).

1.1 Main Window

Jendela Utama adalah jendela terluar Netica. Hal ini terlihat segera setelah Anda masuk ke Netica dan terlihat seperti ini:



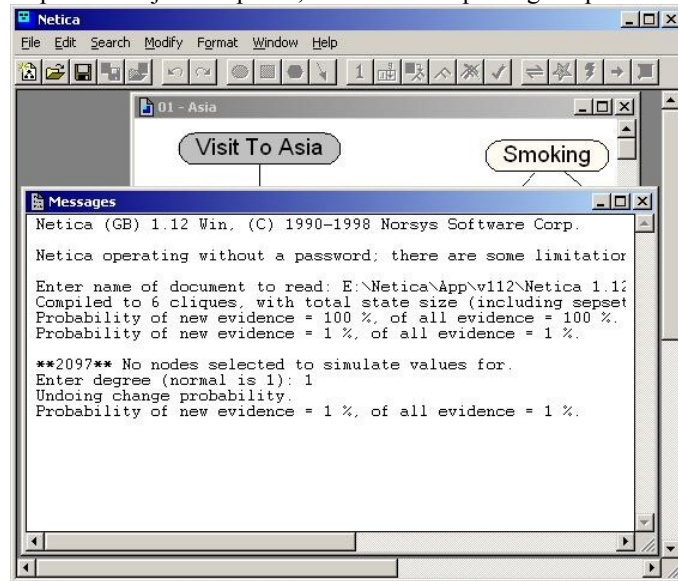
Jendela Utama adalah jendela Microsoft Windows standar yang dapat diubah ukurannya, diminimalkan, ditutup, diseret, dll.

Tombol (dan item menu) yang berwarna abu-abu dinonaktifkan. Mereka menjadi berwarna bila memungkinkan untuk menggunakannya. Saat pertama kali diluncurkan, message window adalah satu-satunya jendela, dan satu-satunya fungsi yang tersedia adalah membuat network baru, membuka network yang ada pada file, atau menyimpan jendela aktif (dalam cases ini, isi Message Window) ke disk.

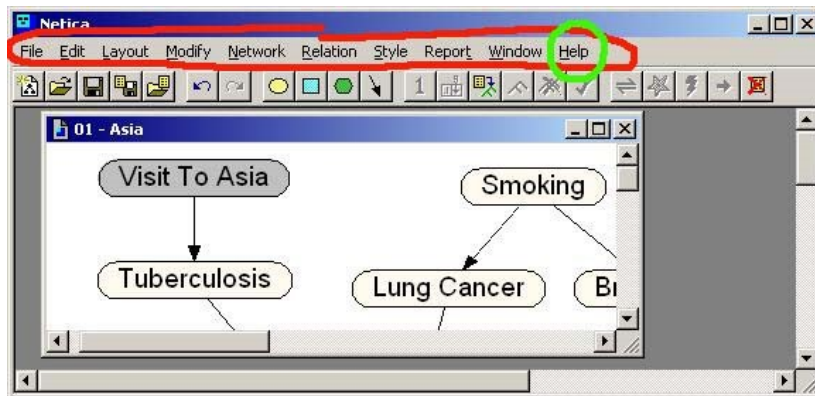
1.2 Message Window

Jendela Pesan awalnya diminimalkan, namun Anda sering ingin membukanya untuk diperiksa, terutama jika terjadi sesuatu yang mengejutkan, atau Netica menampilkan dialog kesalahan. Jendela pesan berisi akun teks langkah-demi-langkah dari semua langkah pemrosesan utama, termasuk detail berguna yang tidak mudah ditampilkan di tempat lain. Saat laporan dibuat, mereka ditulis ke Jendela Pesan.

Berikut adalah contoh snapshot dari jendela pesan, setelah beberapa langkah pemrosesan:



1.3 Menu Bar

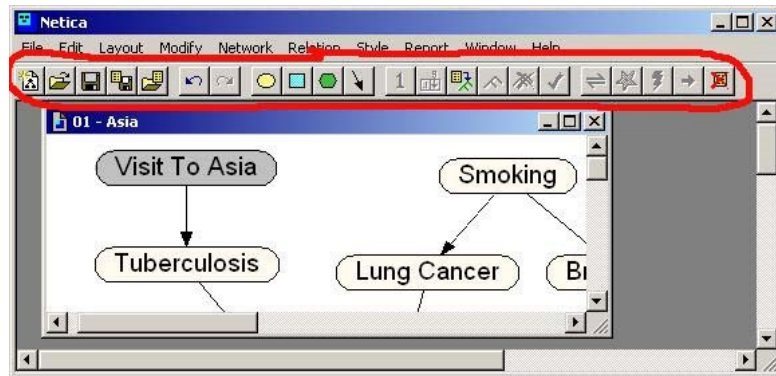


Jelajahi menu bar hanya untuk membiasakan diri dengan banyaknya fungsi yang tersedia. Anda akan belajar sebagian besar fungsi ini melalui sisa tutorial. Cara terbaik adalah mempelajari setiap fungsi dalam konteks nyata, saat Anda termotivasi untuk menggunakannya, jadi kita tidak akan mencoba membahas fungsi menu disini.

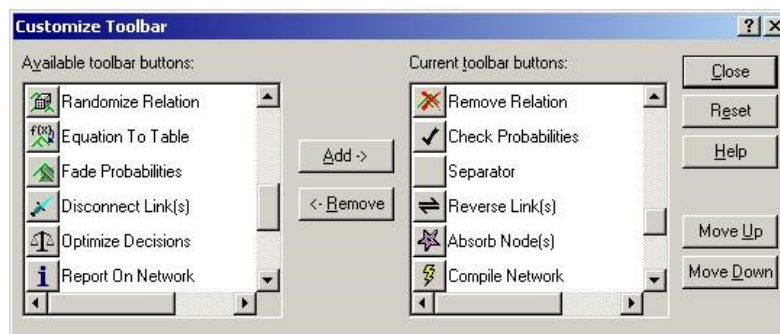
Help System/ Sistem Bantuan

Untuk pengguna Netica awal, mungkin item Menu yang paling penting adalah Help. Netica memiliki sistem bantuan layar yang sangat canggih dan lengkap. Bila Anda memiliki pertanyaan mengenai fungsi tertentu, atau bagaimana mengoperasikan Netica, Anda pasti akan menemukannya di suatu tempat dibawah Help. Cobalah dan gunakan jika sering! Kita di Norsys telah berupaya keras dalam Sistem Bantuan dan sangat bangga karenanya.

1.4 Tool Bar

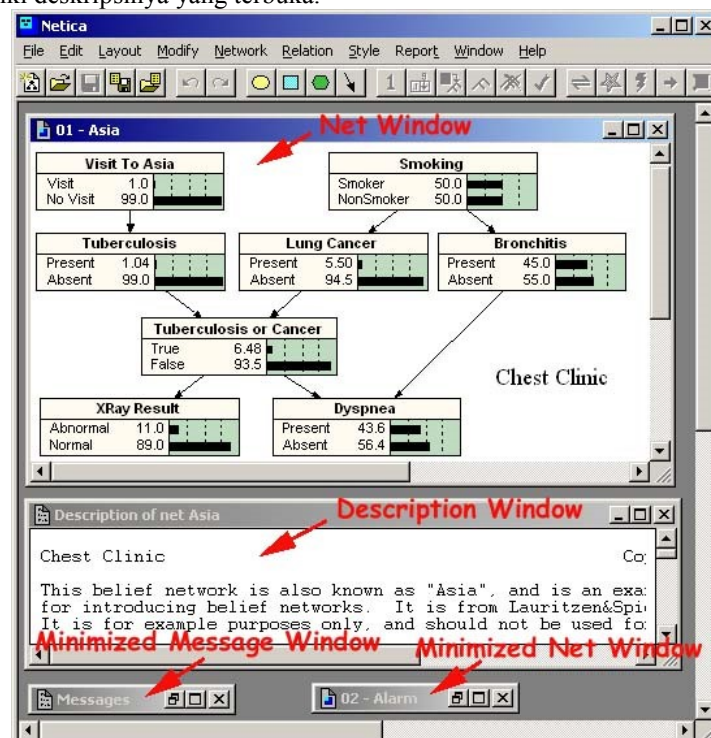


Tombol tool-bar hanya pintasan yang mudah digunakan untuk fungsi yang tersedia pada Menu bar (lihat diatas). Tujuan mereka adalah untuk menghemat klik tombol. Tombol toolbar-bar default adalah aplikasi yang sering digunakan pengguna. Jika Anda ingin menambahkan atau menghapus beberapa tombol, cukup pilih Window → "Customize Toolbar". Kotak dialog Customize Toolbar akan muncul, ditunjukkan disini, dan Anda dapat dengan mudah menambahkan dan menghapus tombol, sesuai keinginan.



1.5 Net Window

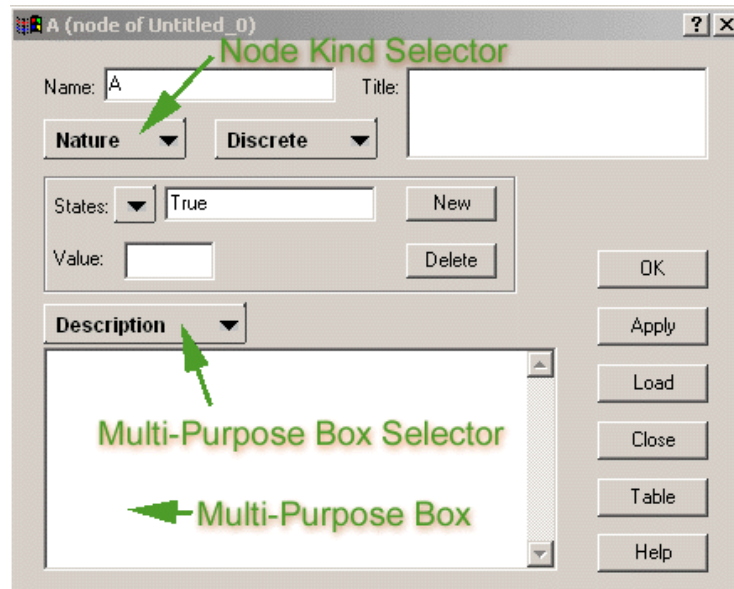
Anda bisa memiliki sejumlah network net. Mereka bisa dibuka atau diminimalkan. Masing-masing dapat dikaitkan dengan jendela teks khusus yang disebut jendela Deskripsi. Dibawah ini kita melihat dua jendela net, salah satunya memiliki deskripsinya yang terbuka.



Ukuran network yang ditampilkan di jendela network dapat dikontrol oleh kontrol zoom yang tersedia di bawah menu "Window". Lihat B.2.9 Mengubah ukuran net dan memperbesar rinciannya. Pilihan font di bawah menu Format juga akan mempengaruhi tampilan ukuran dan tampilan.

1.6 Kotak Dialog Node/ Node Dialog Box

Bila Anda mengklik dua kali pada sembarang node, Node Dialog Box ini akan muncul.



Dari situ Anda bisa mengatur properti dari node itu. Coba tingkatkan kotak dialog node dan jelajahi semua properti yang tersedia untuk disetel atau diubah. Kita akan membahas banyak hal ini di tutorial berikutnya.

2. Membangun Struktur Net

Dalam pendahuluan tutorial, kita melihat bagaimana menggunakan Bayes nets yang sebelumnya telah dibangun untuk Anda. Dalam tutorial ini Anda akan mempelajari langkah-langkah untuk membangun Bayes nets Anda sendiri. Kita berkonsentrasi pada dasar-dasar menganalisis masalah dan menurunkan struktur kausalnya. Ini akan diterjemahkan ke dalam node dan kaitan/ links antara node yang masuk ke Bayes nets. Dalam tutorial selanjutnya kita akan berkonsentrasi pada bagaimana mendefinisikan hubungan probabilistik antara node yang telah dihubungkan.


Langkah pertama dalam membuat network adalah mengklik item menu: File → New Network. Netica akan membuat jendela network kosong untuk Anda. Sebaiknya gunakan perintah 'Save As' segera untuk memberi nama dan simpan network Anda.

2.1 Membuat Node

Langkah selanjutnya adalah menentukan variabel apa yang akan dimodelkan oleh net. Anda kemudian cukup membuat satu node per variabel dan menempatkannya dimanapun Anda inginkan di jendela net, walaupun secara umum, Anda akan menemukannya bekerja dengan baik untuk menempatkan node terkait bersama-sama. Jika ada hubungan kausal antara dua node, konvensi tersebut adalah :

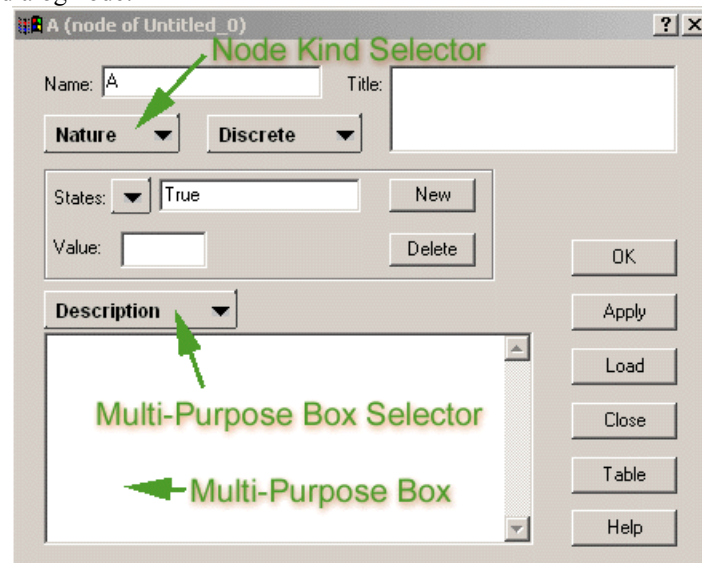
menempatkan node penyebab/ causing node (induk) diatas nodes yang ditimbulkan/ caused node (anak).

Node yang berada di puncak network biasanya dicadangkan untuk penyebab awal/ initial causes, yang kemudian menyebabkan atau mempengaruhi kejadian/ events yang terjadi semakin rendah. Variabel terakhir yang dapat diamati/ diobservasi biasanya ditempatkan dibawah net. Anda membuat *nature node* dengan cara-cara berikut ini:

1. ketik F9
2. klik Modify → Add Nature Node
3. klik  dan seret mouse ke net.

Setelah ini, ikon elips yang jelas akan muncul di jendela net, menunjukkan bahwa sebuah node siap untuk penempatan. Cukup gerakkan mouse ke tempat Anda ingin node itu ditempatkan dan klik lagi. Tentu saja Anda bisa memindahkannya kemudian dengan hanya menyeretnya ke mana saja di jendela net.

Sekarang setelah node itu ada, Anda harus menentukan propertinya. Klik dua kali pada node untuk memunculkan kotak dialog node:



2.1.1 Nama dan Judul/ Titles

Semua node di network harus memiliki nama yang unik. Nama dibatasi hingga 30 karakter atau kurang. Karakter pertama harus berupa huruf. Karakter yang tersisa mungkin berupa huruf, angka (0-9), atau garis bawah ('_'). Nama tidak dapat memiliki spasi atau tanda baca atau karakter lainnya. Nama node digunakan dalam laporan dan pengaturan pembatasan ukuran lainnya.

Saat pertama kali dibuat, node diberi nama default yang unik. Anda dapat mengubah ini ke nama yang valid yang belum pernah diambil oleh beberapa node di net.

Judul/ titles tidak memiliki batasan yang sama dengan namanya; mereka bisa menjadi panjang dan mengandung karakter apapun.

2.1.2 Kind/ Tipe

Mayoritas node di network Anda biasanya adalah *nature nodes*, yang berarti mereka model realitas dibandingkan dengan extra-reality concepts seperti values atau decisions. Node konstan dapat dianggap sebagai tuning/ menyetel parameter; Mereka biasanya dipegang konstan, namun bisa diubah oleh perancang untuk mempengaruhi perilaku net.

Dengan asumsi kita menciptakan sebuah *nature nodes*, sebuah keputusan kunci yang harus dibuat adalah menyatakan apakah node dapat mengambil :

- nilai/ *value* atau keadaan/ *state* yang tak terbatas (*infinite*) atau mempunyai ciri-ciri tersendiri (*discrete*).

Kita akan mengeksplorasi bagaimana membuat pernyataan ini di bagian berikut.

2.1.3 Discrete vs. Continuous

Ingat bahwa masing-masing node (*nature*) di Bayes nets mewakili ciri/ fitur dunia yang menjadi model. Ini adalah properti dari sesuatu yang dapat bervariasi. Variansnya bisa berada diatas himpunan state yang terbatas (*finite set of states*), atau seperangkat nilai yang hampir tak terbatas (*infinite set of values*). Seringkali pembangun Bayes net menggunakan istilah '*node*' dan '*variable*' secara bergantian.

Variabel dapat terdiri dari dua varietas,

- **discrete**, artinya mewakili sejumlah nilai kemungkinan (*possible values*) yang terbatas (misalnya, panas / sedang / dingin, besar / kecil, 1/2/3, dll.), dan
- **continuous**, artinya mewakili kisaran nilai kemungkinan (*range of possible values*) yang tak terbatas (misalnya 0,5-3,4).

Berurusan dengan rentang nilai yang tak terbatas memerlukan pertimbangan khusus untuk teori Bayes net. Hal ini karena banyak algoritma sejauh ini bekerja paling baik bila jumlah state induk yang terbatas dipertimbangkan. Tabel pada dasarnya cocok untuk mewakili sejumlah hal yang berbeda, dan jika Anda tidak menggunakan tabel secara internal, Anda harus menggunakan persamaan dan memiliki mesin matematika simbolis yang sangat canggih untuk menghitung batasan di antara semua persamaan.

Netica dapat menangani variabel *continuous* tanpa kesulitan. Namun, karena semua inferensi/ kesimpulan probabilitasnya dilakukan dengan tabel diskrit, secara internal harus mengubah variabel *continuous* menjadi jumlah/ kuantitas *discrete* sebelum ada inferensi/ kesimpulan probabilitas yang dapat dilakukan.

Kotak dialog node memiliki **Discrete** selector, yang memungkinkan Anda memilih apakah nodes mewakili variabel *continuous* atau *discrete*. Pilih state mana yang tepat untuk node tertentu itu.

Perlu dicatat bahwa meskipun sebuah variabel berperilaku dalam satu mode dominan, seperti *discrete* atau *continuous*, biasanya kita ingin memperlakukannya seolah-olah berperilaku berbeda. Ini mudah dilakukan dengan Netica, dan karena itu Anda tidak berkomitmen untuk memperlakukan variabel *continuous* seperti semata-mata *continuous*, atau sebaliknya.

Misalnya, ambil voltase baterai. Anda mungkin biasanya menganggap tegangan sebagai kuantitas *continuous*. Namun, terkadang, mungkin berguna untuk menentukan/ *discretize* ukuran tegangan menjadi rentang nilai. Battery's charge bisa dianggap lemah, sedang, atau kuat (*weak*, *medium*, atau *strong*), tergantung pada apakah voltasenya jatuh dalam kisaran, 0-0,5, 0,5-5, atau 5,0 dan lebih tinggi. Demikian juga, ada kalanya variabel *discrete*, katakanlah salah satu yang Benar atau Salah (*True* atau *False*), dapat dengan mudah dikaitkan dengan nilai numerik. Katakan True adalah tegangan 3.0, sedangkan False adalah 2.0.

Untuk mempelajari bagaimana menetapkan variabel *continuous* suatu himpunan rentang *discrete*, atau untuk memberikan keadaan/ state dari sejumlah variabel *discrete*, lihat "Node Ranges" pada sistem bantuan/ help system di layar.

Catatan: Sebuah *utility node* harus *continuous* dan *decision node* harus *discrete*.

2.1.4 States

Secara *default*, sebuah node dibuat dengan dua state, *True* dan *False*. Anda bisa memberi nama state lain dengan terlebih dahulu memilih keadaan/ state yang diinginkan, lalu memasukkan namanya di jendela. Field "*Value*" adalah untuk mengaitkan nilai bilangan real dengan keadaan/ state yang dipilih. Tombol "*New*" adalah untuk menambahkan state.

Jika Anda memiliki banyak state (nama dan / atau nilai) untuk dimasukkan, atau Anda ingin menghapus state, Anda akan ingin menggunakan *Multi-Purpose box*. Untuk menetapkan nama state, pilih "*State*". Untuk memilih nilai state, pilih "*Range*". Lihat bantuan/ help di layar untuk sintaks baris yang diterima oleh *Multi-Purpose box*.

2.1.5 Deskripsi dan Properti Lainnya

Node dapat diberikan banyak string dokumentasi yang berguna, termasuk Description or Comment field, Author field, dan bahkan arbitrary name-value pairs/ pasangan nama-nama yang berubah-ubah. Lookup "*Multi-Purpose Box*" dalam bantuan/ help di layar untuk rinciannya.

2.1.6 Pembuatan Node Cepat/ Rapid Node Creation

Jika Anda perlu membuat banyak node dari tipe tertentu, cukup klik dua kali pada ikon tool bar untuk tipe node tersebut:



untuk Nature nodes,



untuk Utility nodes,



untuk Decision nodes


Ini akan menempatkan Netica dalam "rapid node creation mode". Setelah dalam mode ini, setiap kali Anda mengklik di jendela net, sebuah node dari tipe itu akan dibuat di lokasi itu. Untuk meninggalkan "rapid node creation mode", cukup klik satu tombol pada tool bar.

2.1.7 Membuat Node Secara Otomatis Dari Case files/ Cases Files


Sekarang setelah Anda belajar membuat node secara manual, Anda akan senang mengetahui bahwa Netica juga dapat membuat node secara langsung dari kumpulan data. Ini adalah fitur menghemat waktu yang bagus, asalkan Anda memiliki kumpulan data yang terdefinisi dengan baik. Untuk memanfaatkan fitur ini, dataset harus berada dalam *Case File Format*.

2.2 Menambahkan Link

Setelah Anda membuat dua atau lebih node, Anda mungkin mulai menghubungkannya bersama-sama. Anda membuat link dengan cara-cara berikut ini:

- ketik F12
- klik Modify → Add Link
- klik 

Ini akan memberi kursor berupa panah. Selanjutnya klik pertama pada node induk dan kemudian node anak. Panah akan diambil dari induk kepada anak, yang menunjukkan bahwa mereka sekarang terhubung. Memindahkan dan menghapus link dibahas di Bagian _ dan _.

Rapid Link Creation Mode: Sama seperti "rapid node creation mode", ada pilihan yang mudah digunakan didalam Netica yang memungkinkan Anda memasukkan "rapid link creation mode" untuk menghemat klik mouse secara ekstra. Untuk masuk ke mode ini, cukup klik dua kali pada ikon tool bar . Setelah itu, setiap kali Anda mengklik sepasang node, sebuah link akan dibuat antara pada pasangan yang pertama ke pasangan kedua. Untuk meninggalkan "rapid link creation mode", cukup klik satu klik pada ikon tool bar apa pun.

2.3 Memindahkan Node dan AutoGrid

Untuk memindahkan sebuah node, klik di bawahnya, seret ke posisi barunya, lalu lepaskan tombol mouse. Jika Anda ingin memindahkan sekumpulan node, seret kotak di sekitar kumpulan nodes untuk dipilih, lalu klik salah satu di antaranya dan seret seluruh rangkaian ke lokasi yang diinginkan. Anda dapat menambahkan ke kumpulan yang telah dipilih sebelumnya dengan menekan tombol ctrl atau shift saat memilih, yang merupakan cara standar yang digunakan File Managers dalam Microsoft Windows atau Macintosh OS. Oleh karena itu, jika Anda terbiasa dengan sistem tersebut, Anda akan menemukan objek yang dipilih dan bergerak didalam Netica sangat alami.

Bila Anda memindahkan node, semua link yang mereka miliki akan bergerak secara otomatis bersamaan dengan mereka.

AutoGrid: Ada grid posisi/ underlying yang tidak terlihat dan tak terlihat di jendela net, dan saat Anda menambahkan atau memindahkan node, node akan sedikit bergeser sehingga pusatnya berada tepat di atas salah satu posisi grid ini, sehingga AutoGrid dihidupkan. Ini memungkinkan Anda untuk dengan cepat menggambar network yang nodenya selaras sempurna, dan dengan link yang berjalan sempurna horizontal atau vertikal sempurna. Untuk mematikan atau mengaktifkan AutoGrid, toggle Layout → AutoGrid. Jika Anda ingin mengubah jarak grid, untuk membuatnya lebih baik atau courser, pilih Layout → Grid-Spacing, dan sebuah kotak dialog akan muncul memungkinkan Anda membuat setting.

Nudging: Terkadang berguna untuk bisa memindahkan node, atau set node, jumlah yang dapat diprediksi kecil. Anda bisa menggunakan tombol panah untuk melakukan penyambungan ini. Cukup pilih node yang ingin Anda pindahkan, dan kemudian tekan tombol dengan panah ke arah yang Anda inginkan untuk dipindahkan. Jarak yang akan ditempuh dengan masing-masing menekan tombol adalah jarak grid (menyediakan AutoGrid dihidupkan), sehingga Anda dapat menyesuaikan jumlah ini dengan menggunakan entri menu Layout → Grid-Spacing. Jika AutoGrid dimatikan, maka dengan setiap menekan tombol mereka akan dipindahkan jumlah yang paling kecil.

2.4 Memilih Node dan Links

Seleksi merupakan operasi penting di Netica, karena banyak operasi dilakukan dengan terlebih dahulu memilih satu atau beberapa item, dan kemudian memilih operasi untuk dilakukan pada mereka. Netica menawarkan banyak operasi berguna untuk menangani objek yang dipilih, termasuk menciptakan, menyimpan, dan mengembalikannya/ creating, saving, and restoring.

Netica menunjukkan objek yang dipilih dengan menggambarnya hilited atau dengan warna negatif.

2.4.1 Node

Pilih sebuah node dengan mengklik sekali di atasnya. Untuk memilih sekelompok node, klik pada latar belakang dan kemudian tarik persegi panjang pilihan untuk memasukkan setidaknya satu bagian dari masing-masing. Untuk membatalkan pilihan semua node yang saat ini dipilih, cukup klik sekali pada latar belakang di jendela. Memilih Edit → Select-All (atau mengetikkan CTRL + A, seperti konvensi Windows) akan memilih semua node dalam network.

Memodifikasi: Bila Anda memilih node baru, setiap nodes yang sebelumnya dipilih akan tidak dipilih, kecuali Anda menekan tombol CTRL atau SHIFT saat mengklik atau menyeret. Dalam hal ini status terpilih dari node yang diberikan akan dibalik. Ini memungkinkan Anda menambahkan atau menghapus dari koleksi node terpilih saat ini.

Reversing/ Membalikkan: Untuk membalikkan pemilihan sehingga hanya mencakup node yang saat ini tidak dipilih, tahan tombol SHIFT saat memilih Edit → Select-All. (Pada rilis Netica berikutnya, item menu terpisah, "Edit → Select Nodes → Invert Selection", disediakan untuk ini.)

Dari Teks: Edit → "Select Copied Nodes" akan memilih semua node yang namanya (bukan title) muncul pada teks yang saat ini disalin ke clipboard. Misalnya, jika program pengolah kata Anda memiliki dokumen yang menyebutkan beberapa node menurut nama, Anda bisa memilih teks itu, menyalinnya, dan kemudian melakukan "Choose Copied Nodes" di Netica untuk memilihnya.

Save/ Restore: Anda dapat menyimpan pilihan saat ini sebagai teks dengan Report → "List Selected Node", dan kemudian mengembalikannya menggunakan Edit → "Select Copied Nodes".

Tidak Bekerja: Jika Anda memilih node gaya belief-bar atau beief-meter, maka Anda harus mengeklik title (atau nama) nodes, karena mengeklik di tempat lain dapat ditafsirkan saat memasukkan temuan/ findings (dalam hal ini node akan menjadi gelap bukan digambar dengan warna negatif).

2.4.2 Link


Untuk memilih link, klik kiri di atasnya. Ini akan ditarik dalam bentuk garis besar. Sekali lagi, jika tombol CTRL ditekan saat mengklik, ia akan membalikkan status pemilihan hanya link yang diklik, yang dapat digunakan untuk menambahkan atau menghapus dari kumpulan link yang dipilih. Link tidak dapat dipilih dengan menyeret persegi panjang pilihan di atas mereka. Node dan link tidak dapat dipilih pada saat bersamaan, jadi kapan pun Anda memilih link, node apa pun yang dipilih akan menjadi tidak dipilih, dan sebaliknya.

2.5 Membatalkan & Mengulangi/ Undoing & Redoing

Saat membangun atau mengubah network Anda dapat membatalkan operasi terakhir dengan salah satu dari:

- pilih Edit → Undo
- mengetikkan CTRL + Z
- klik 

dan Anda dapat mengulang operasi dengan salah satu dari:

- pilih Edit → Redo
- mengetikkan CTRL + SHIFT + Z
- klik 

Anda dapat membatalkan operasi sebanyak yang diketahui Netica, yang bervariasi, bergantung pada jumlah memori yang dibutuhkan untuk melacak keadaan sebelumnya. Netica akan selalu mengingat setidaknya empat operasi. Bila Anda menyimpan network, Netica menjalankan ingatannya; Oleh karena itu, Anda tidak bisa lagi mengurungkan operasi sebelum disimpan.

Saat item menu Edit → Undo atau tombol toolbar berubah redup, Anda kehabisan undos.

Saat item menu Edit → Redo atau tombol toolbar berubah redup, Anda telah memulihkan semua operasi yang sebelumnya dibatalkan.

Operasi yang dilakukan di setiap jendela akan diingat secara terpisah, jadi jika Anda kembali ke jendela A setelah bekerja di jendela lain untuk sementara, Anda masih dapat membatalkan operasi yang sebelumnya dilakukan di jendela A.

Menjadi akrab dan berpengalaman lebih awal dengan Undo dan Redo, karena mereka sangat berguna dan menghemat waktu operasi.

2.6 Reshaping/ Membentuk kembali Link

Bila diagram network menjadi besar, sangat sulit untuk melihat apakah semua hubungan antara nodes adalah garis lurus. Bahkan diagram kecil terkadang bisa dibuat lebih mudah dibaca dengan memilih jalur yang sesuai untuk link.



Menambahkan Bends: Untuk membentuk sebuah link, pertama klik sekali pada link untuk memilihnya. Link akan ditarik dengan warna hilite yang mengelilinginya. Kemudian klik lagi pada link yang dipilih, dan seret kursor. Sebuah tikungan akan ditempatkan di link dan diseret oleh kursor. Anda bisa mengulanginya untuk menambahkan banyak tikungan sesuai keinginan. Cobalah ini untuk melihat betapa mudah dan alami itu.

Memindahkan Bends: Untuk menggerakkan posisi tikungan, pertama pilih link seperti sebelumnya. Link akan menjadi hilited dan akan ada kotak persegi di setiap tikungannya, dan pada dua titik akhirnya. Untuk memindahkan tikungan atau titik akhir, klik di kotaknya yang hilited, seret ke posisi barunya, lalu lepaskan tombol mouse. Memindahkan satu tikungan ke yang lain, atau ke titik akhir, akan menggabungkannya, yang bisa digunakan untuk menghilangkan lengkungan. Coba gerakkan tikungan yang Anda buat di atas. Sekarang coba gabungkan ke salah satu titik akhir untuk menghapusnya.

Moving End: Ada beberapa pertimbangan khusus saat memindahkan titik akhir sebuah link. Ujung panah dari sebuah link tidak dapat diseret sangat jauh dari node yang ditunjukkannya (tanda panah hanya akan bertemu dengan penghalang tak terlihat). Ini mencegah pembuatan diagram yang menyedatkan atau membingungkan (karena link masih dianggap terhubung ke nodes). Jika Anda menyeret ujung panah non-panah yang terlalu jauh dari node induknya, link akan menjadi "terputus" dari nodes tersebut. Anda bisa tahu ini terjadi karena nama linknya tiba-tiba muncul di endpoint yang telah Anda pindahkan. Lihat bantuan di layar untuk informasi lebih lanjut tentang subjek ini.


AutoGrid dan Nudging: Sama seperti dengan node, fitur AutoGrid berlaku untuk link, memungkinkan Anda membuat link yang sangat sejajar dan diposisikan dengan baik.

2.7 Saving and Restoring/ Menyimpan dan Memulihkan Network

Saving: Untuk menyimpan network aktif ke file, pilih File → Save As atau klik . Jika file belum disimpan sebelumnya, Anda akan diminta memasukkan nama file dan direktori apa yang akan dimasukkan ke dalamnya. Secara default, file tersebut diberi akhiran ".dne"). Jika item menu File → Save, atau  tombolnya redup, ini merupakan indikasi bahwa Menyimpan dinonaktifkan karena network belum pernah berubah sejak disimpan atau dibaca terakhir.

Menyimpan dengan atau tanpa Findings: Jika ada findings (evidence) yang dimasukan didalam net, mereka akan disaved juga. Jika Anda tidak menginginkan ini, pilih Network → "Remove Findings" sebelum menyimpannya.

File Format: Network disimpan ke file yang hanya terdiri dari teks ASCII, sehingga mudah ditransfer dari satu tipe komputer ke komputer lain (misalnya dari Windows ke Unix ke Macintosh) atau dikirim melalui email. Anda bisa memeriksa dan memodifikasi file dengan editor teks. Ini ada dalam file DNET-1 format.


Restoring: Untuk membaca network yang tersimpan sebelumnya dan menjadikannya network aktif, pilih File → Open ... atau klik .

Cobalah menyimpan dan memulihkan network Anda saat ini. Sebaiknya Anda terbiasa membuat banyak "pos pemeriksaan/ checkpoint" versi network Anda, saat Anda membangunnya, jika Anda ingin kembali ke versi sebelumnya yang tidak dapat dipulihkan melalui fitur "Undo". Selain itu, Anda sering dapat menghemat banyak waktu dengan mengedit net orang lain yang sebelumnya dibangun (di Netica, atau dengan menggunakan editor teks pada file .dne) alih-alih membuat sendiri dari nol. Banyak pengguna bahkan memanfaatkan program untuk secara otomatis membuat file .dne, yang kemudian dapat dibaca oleh Netica.

2.8 Menghapus Node dan Link

2.8.1 Node

Untuk menghapus node atau nodes, pertama pilih masing-masing node tertentu, dan kemudian lakukan salah satu dari:

- tekan tombol DELETE
- pilih Edit → Hapus
- klik 

Ini akan menghapus node dan semua link dari node lain ke mereka. Link yang keluar dari mereka ke node anak mereka akan terputus sesaat sebelum penghapusan. Link yang terputus ini akan dipilih setelah node dihapus, sehingga Anda dapat dengan mudah menghapusnya juga dengan hanya menekan DELETE untuk kedua kalinya.

Cobalah membangun network dengan rantai tiga node dan kemudian hapus node tengah dan link yang terputus yang kemudian terbentuk.

Jika Anda ingin menghapus node, namun menjaga hubungan global dari node yang tersisa (yaitu, *joint distribution*), Anda harus menyerap/ *absorb* [untuk dihubungkan/ to be linked ...] nodes. Ini adalah operasi yang sangat hebat yang dapat digunakan untuk menyederhanakan network, jika Anda menemukan alasan apapun mengapa Anda memiliki terlalu banyak node.

2.8.2 Link

Untuk menghapus sebuah link, klik di atasnya untuk memilihnya, dan kemudian melakukan penghapusan, seperti di atas. Untuk menghapus serangkaian link, pilih semuanya dan kemudian lakukan penghapusan. Ketika sebuah link dihapus, *conditional probabilities* dari nodes anak runtuh untuk menghilangkan ketergantungan mereka pada induk itu, seolah-olah induk mengambil state pertamanya. Ini mungkin atau mungkin tidak diinginkan, jadi Anda mungkin ingin meninjau kembali *conditional probability tables* anak sebelumnya setelah melakukan langkah seperti itu.

Bila link dihapus, nodes induk tidak terpengaruh dengan cara apa pun.

2.9 Mengubah Ukuran Net dan Memperbesar/ Net Resizing and Zooming

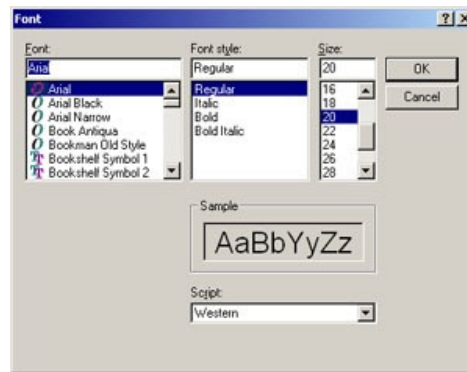
Bila diagram network menjadi besar, sangat berguna untuk dapat memperkecil tampilan lebih banyak di jendela sekaligus. Selain itu, saat mempersiapkan network untuk presentasi grafis, Anda mungkin ingin mengukurnya dengan tepat. Netica menawarkan beberapa cara mudah untuk menyesuaikan ukuran net.

Perintah zoom lengkap yang tersedia dari menu Window → Zoom atau keyboardnya adalah sebagai berikut:

| Zoom Menu | Keyboard | Description |
|-----------|--------------|--|
| In | CTRL+> | Zoom's In. |
| Out | CTRL+< | Zoom's Out. |
| To Fit | CTRL+SHIFT+> | Zooms just enough so that the whole network will be displayed flush with the window borders, with a small margin/ Zoom cukup sehingga seluruh network akan ditampilkan flush dengan batas jendela, dengan margin kecil |
| To Normal | CTRL+SHIFT+< | Return to 100% magnification/ Kembali ke pembesaran 100% |
| To... | CTRL+ALT+> | Allows you to specify an exact magnification amount/ Memungkinkan Anda menentukan jumlah pembesaran yang pasti |
| Back | CTRL+ALT+< | Returns the magnification to what it was before the last zoom operation/ Mengembalikan pembesaran ke apa yang terjadi sebelum operasi zoom terakhir |

Percobaan dengan setiap operasi zoom, karena semuanya sangat berguna.

Nodes: Node dalam network dapat diubah ukurannya dengan menyesuaikan font teks mereka. Pilih item menu Style → Font untuk menampilkan dialog pemilihan dengan fitur lengkap.




Catatan: Perbesaran khusus yang digunakan untuk menampilkan network, atau nodesnya, tidak mempengaruhi operasi yang dilakukan pada network.

3. Mendefinisikan Hubungan Node/ Defining Node Relationships

Pada tutorial sebelumnya kita melihat bagaimana membangun struktur dasar network, yaitu bagaimana menentukan node dan menghubungkannya. Di sini kita akan belajar bagaimana menentukan hubungan probabilitas antara node yang telah dihubungkan.

Meskipun Anda mungkin berpikir bahwa hubungan secara alami akan menjalin hubungan ini, ternyata ini tidak ideal, karena ini membuat sulit untuk menentukan saling ketergantungan/ *interdependence* antara hubungan/ *relationships*. Ternyata yang terbaik jika node *holds the relationships*/ memegang hubungan yang dihadapinya dengan induknya. Oleh karena itu, Anda akan menemukan probabilitas bersyarat/ conditional probabilities yang terkait dengan link apa pun dengan memeriksa nodes anak dari link tersebut. Anda dapat menemukan hubungan dengan baik:

1. pertama pilih node anak dan pilih Relation → View Edit, atau
2. pertama pilih node anak dan kemudian klik 
3. memunculkan kotak dialog node anak, dan mengklik tombol "Table".

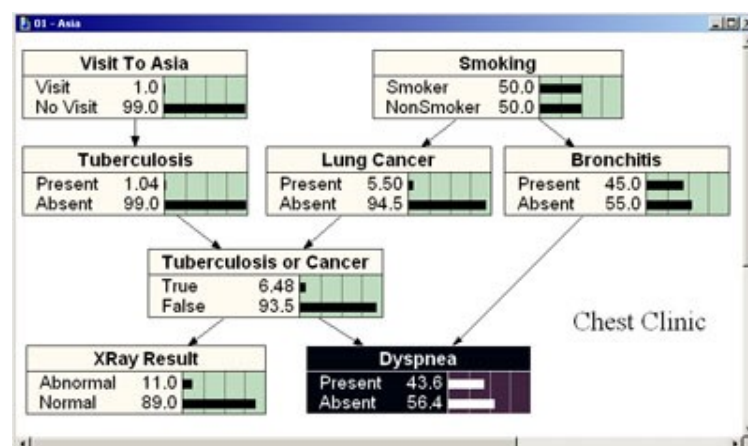
3.1 Mendefinisikan Tabel Probabilitas secara Manual

Cara yang paling mendasar dan langsung untuk menentukan *conditional probability* antara node dan induknya adalah secara eksplisit menentukan apa yang disebut Conditional Probability Table, atau CPT, singkatnya.

CPT hanyalah sebuah tabel yang memiliki satu kemungkinan/ probabilitas untuk setiap kemungkinan kombinasi antara keadaan/ state induk dan anak. Ini adalah tabel dimensi $N + 1$, dimana N adalah jumlah induk.

Namun tabel dapat "diratakan/ flattened" menjadi dua dimensi dengan secara eksplisit menentukan semua kombinasi keadaan/ state induk dalam satu dimensi dan semua keadaan/ state anak didimensi lainnya.

Inilah yang dilakukan Netica, karena tabel multidimensi sulit divisualisasikan. Mari kita lihat CPT untuk nodes Dyspnea didalam net Asia:



| TbOrCa | Bronchitis | Present | Absent |
|--------|------------|---------|--------|
| True | Present | 90.000 | 10.000 |
| True | Absent | 70.000 | 30.000 |
| False | Present | 80.000 | 20.000 |
| False | Absent | 10.000 | 90.000 |

Dyspnea memiliki dua induk, TbOrCa dan Bronchitis. Masing-masing memiliki dua state. Disebelah kiri, disajikan secara vertikal, semua kemungkinan kombinasi dari keadaan induk. Dikanan atas semua keadaan Dyspnea, "Present" dan "Absent". Probabilitas setiap kombinasi keadaan induk dan keadaan anak kemudian diberikan dikanan bawah.

Rows must sum to 1.0/ Baris harus berjumlah 1.0. Perhatikan bahwa probabilitas setiap baris dalam tabel harus berjumlah persis sampai 1.0. Ini karena setiap baris meringkas probabilitas satu kemungkinan dunia, satu dimana induk berada distate yang ditentukan. Dan untuk kemungkinan dunia itu, kemungkinan/ chance dari anak berada dalam keadaan apapun harus berjumlah 1,0.

Anda dapat mengedit probabilitas dalam tabel hanya dengan mengklik pada kolom kolom yang sesuai dan mengetikkan nilai baru. Setelah selesai, klik "Oke". Jika salah satu baris tidak berjumlah 1.0, muncul dialog kesalahan; cukup buat koreksi yang diperlukan dan kemudian klik "Oke" lagi.

Terminology: Deterministic vs. Probabilistic relations. Terkadang node anak hanya memiliki satu nilai yang mungkin untuk setiap kemungkinan konfigurasi keadaan induk. Node seperti itu dikatakan *deterministic* karena nilainya ditentukan persis oleh induknya, tidak ada unsur kebetulan/ chance yang terlibat. Ini berarti bahwa di CPT anak, disetiap baris, satu kolom akan memiliki nilai 1,0, dan semua kolom lainnya akan memiliki nilai 0,0.

Load Button/ Tombol Load: Mengklik tombol Load akan memperbarui semua probabilitas dalam tabel untuk mencerminkan keadaan saat ini di network. Ini adalah kebalikan dari tombol "Apply" yang menyalin tabel ke dalam network. Fitur "Load" bukan yang banyak digunakan oleh pengguna pemula, namun bila Anda melakukan analisis "bagaimana jika/ what if" yang kompleks, itu bisa sangat berguna (sangat baik didalam Netica agar dialog beberapa tabel terbuka untuk node yang sama, masing-masing dengan probabilitas yang berbeda yang Anda bereksperimen dengannya). Juga, terkadang network yang mendasari diperbarui oleh sebuah program (katakanlah kapan net mempelajari/ learning probabilitasnya dari cases yang terjadi saat berjalan), dan Anda ingin mengikuti arus probabilitas saat ini, jadi Anda bisa terus mengklik "Load" untuk melihat probabilitas terbaru.

3.2 Mendefinisikan Tabel Probabilitas dengan Persamaan

Tabel terkadang bisa merepotkan/ tidak praktis untuk memasukan dengan tangan, terutama jika ada banyak state induk untuk dipertimbangkan. Netica menawarkan kemampuan untuk membuat deskripsi singkat tentang *conditional probability tables* dengan menggunakan persamaan. Bahasa persamaannya lengkap dan kuat, dan mengikuti sintaks bahasa pemrograman yang populer C, C ++, dan Java.

Tabel dapat digunakan apakah node bersifat *continuous* atau *discrete*, dan apakah relasinya bersifat *probabilistic* atau *deterministic*. Semua persamaan harus dikonversi ke tabel sebelum menyusun network, melakukan transformasi network atau memecahkan masalah keputusan. Tabel kemudian digunakan dengan cara yang sama seperti jika Anda memasukkannya dengan tangan. Karena tabel mengasumsikan kumpulan/ *discrete set* dari state pada bagian dari induk dan anak, setiap node *continuous* yang mengambil bagian dalam sebuah persamaan harus *discretized* terlebih dahulu.

Kita akan belajar bagaimana menggunakan persamaan dengan mempelajari bentuk sintaksis dasar mereka, dan kemudian dengan melihat beberapa contohnya. Anda kemudian harus mencoba untuk membuat beberapa untuk diri sendiri.

Sintaks untuk persamaan sedikit berbeda tergantung pada apakah nilai node ditentukan secara determinatif/ *deterministically* oleh induknya (selalu memiliki nilai unik, untuk setiap konfigurasi keadaan/ state induk) atau ditentukan secara probabilistik/ *probabilistically*.

Deterministic nodes

Sintaks:

Child(Parent1, Parent2, ... Parent N) = beberapa ekspresi yang menghasilkan nilai state legal dari Child

Contoh:

/ Mengkonversi F ke Centigrade */*

$C(F) = 9.0 / 5 * (F - 32)$

/ Total jarak tempuh, X, adalah*

*rata-rata kecepatan (velocity) * waktu tempuh (time traveled) + jarak awal (initial distance) */*

$X(\text{Vel}, dt, X0) = X0 + \text{Vel} * dt$

/ if rasa asam, pilih warna biru; else*

if rasa manis, pilih merah; else

if rasa asin, pilih hijau; else

pilih abu-abu;

**/*

Warna(Rasa) = Rasa == asam? biru:

Rasa == manis? merah:

Rasa == asin? hijau: abu-abu

Probabilistic nodes

Sintaks:

p(Child | Parent1, Parent2, ... Parent N) = beberapa ekspresi yang menghasilkan probabilitas, yaitu angka dalam kisaran 0,0 sampai 1,0.

Contoh:

/ Total jarak tempuh, X, mengikuti distribusi normal dengan rata-rata Vel * dt + X0, dan standar deviasi 'spread' */*

$p(X | \text{Vel}, dt, X0, \text{spread}) = \text{NormalDist}(X, \text{Vel} * dt + X0, \text{spread})$

/ Warna kimiawi adalah fungsi probabilistik dari suhu:*

if suhunya tinggi, warnanya selalu kuning;

if suhu sedang, warnanya selalu oranye;

*Tapi if suhunya rendah, warnanya bisa oranye 20% dari waktu dan merah 80% dari waktu. */*

$p(\text{Warna} | \text{Suhu}) = \text{Suhu} == \text{tinggi?} (\text{Warna} == \text{kuning? } 1.0: 0.0):$

$\text{Suhu} == \text{sedang?} (\text{Warna} == \text{orange? } 1.0: 0.0):$

$\text{Suhu} == \text{rendah?} (\text{Warna} == \text{orange? } 0.2:$

$\text{Warna} == \text{merah? } 0.8: 0.0): 0$

Perhatikan bahwa spasi dan carriage return tidak memiliki bantalan apapun dalam sintaks persamaan; Anda bisa menggunakan sebanyak atau sesedikit yang Anda suka, sesuai selera Anda. Anda juga bisa menambahkan komentar bergaya C (*/ * ... * /*) di manapun Anda suka.

Terdapat beberapa aturan yang harus diikuti agar sebuah persamaan masuk akal bagi Netica:

1. Satu-satunya node yang dapat disebutkan dalam sebuah persamaan adalah: node yang dideskripsikan oleh persamaan, induk, dan node konstan.
2. Jika persamaannya adalah untuk node probabilistik, sisi kanannya harus memberikan probabilitas **semua** nilai node yang mungkin. Anda tidak dapat meninggalkan apapun, karena jika tidak, tabel probabilitas tidak lengkap didefinisikan. Ada satu pengecualian untuk aturan ini. Dalam cases node boolean (yang mengambil nilai true dan false), Netica akan berasumsi bahwa jika Anda hanya memberi satu probabilitas, itu adalah untuk keadaan True.
3. Ini adalah kesalahan umum bagi pengguna awal dari persamaan untuk menentukan fungsi probabilistik yang tidak menghasilkan angka, namun dinyatakan. Misalnya, mereka mungkin mengetik:

p(Child | Parent) = (Parent == True)?Flase:True, tapi Flase dan True bukan probabilitasnya, jadi nilai yang tidak tepat/ tida legal untuk sisi kanan dari fungsi probabilistik. Persamaan yang benar adalah:



p(Child | Parent) = (Parent == True)? 0.0: 1.0 , dan **p(Child | Parent) = not(Parent)** akan bekerja dengan baik.

Bantuan/ help dilayar Netica berisi daftar lengkap semua fungsi persamaan yang ada, termasuk manual referensi terperinci yang menjelaskan parameter dan fungsi sebenarnya mereka. Ada lebih dari 50 diantaranya, dari matematika sederhana hingga statistik kompleks. Berikut adalah daftar mereka dengan nama, hanya agar Anda memiliki gagasan tentang apa yang tersedia.

| Common Functional | Operators Notation | Common Operator | Operators Notation | Common | Math | Special | Math |
|-----------------------|--------------------|-----------------------|--------------------|----------|--------|---------------|-------------------------|
| neg | (x) | - | x | abs | (x) | round | (x) |
| not | (b) | ! | | sqrt | (x) | roundto | (dx, x) |
| equal | (x, y) | x == | y | exp | (x) | approx_eq | (x, y) |
| not_equal | (x, y) | x != | y | log | (x) | eqnear | (reldiff, x, y) |
| approx_eq | (x, y) | x ~ | y | log2 | (x) | clip | (min, max, x) |
| less | (x, y) | x < | y | log10 | (x) | sign | (x) |
| greater | (x, y) | x > | y | sin | (x) | xor | (b1, b2, ... bn) |
| less_eq | (x, y) | x <= | y | cos | (x) | increasing | (x1, x2, ... xn) |
| greater_eq | (x, y) | x >= | y | tan | (x) | nondecreasing | (x1, x2, ... xn) |
| plus | (x1, x2, ... xn) | x1 + x2 + ... + xn | | asin | (x) | min | (x1, x2, ... xn) |
| minus | (x, y) | x - y | | acos | (x) | max | (x1, x2, ... xn) |
| mult | (x1, x2, ... xn) | x1 * x2 * ... * xn | | atan | (x) | argmin0/1 | (x0, x1, ... xn) |
| div | (x, y) | x / y | | atan2 | (x, y) | argmax0/1 | (x0, x1, ... xn) |
| mod | (x, base) | x % base | | sinh | (x) | nearest0/1 | (val, c0, c1, ... cn) |
| power | (x, y) | x ^ y | | cosh | (x) | select0/1 | (index, c0, c1, ... cn) |
| and | (b1, b2, ... bn) | b1 && b2 && ... && bn | | tanh | (x) | member | (elem, s1, s2, ... sn) |
| or | (b1, b2, ... bn) | b1 b2 ... bn | | floor | (x) | factorial | (n) |
| if (test, tval, fval) | | test ? tval : fval | | ceil | (x) | logfactorial | (n) |
| | | | | integer | (x) | gamma | (x) |
| | | | | frac (x) | | loggamma | (x) |
| | | | | | | beta | (z, w) |
| | | | | | | erf | (x) |
| | | | | | | erfc | (x) |
| | | | | | | binomial | (n, k) |
| | | | | | | multinomial | (n1, n2, ... nn) |

| Continuous | Probability | Distributions | Discrete | Probability | Distributions |
|------------------|-----------------|---------------|--------------------|---------------------------------------|---------------|
| UniformDist | (x, a, b) | | SingleDist | (k, c) | |
| TriangularDist | (x, a, w) | | DiscUniformDist | (k, a, b) | |
| NormalDist | (x, m, s) | | BernoulliDist | (k, p) | |
| LognormalDist | (x, h, f) | | BinomialDist | (k, n, p) | |
| ChiSquareDist | (x, n) | | PoissonDist | (k, m) | |
| ExponentialDist | (x, l) | | HypergeometricDist | (k, n, s, N) | |
| GammaDist | (x, a, b) | | NegBinomialDist | (k, n, p) | |
| WeibullDist | (x, a, b) | | GeometricDist | (k, p) | |
| BetaDist | (x, a, b) | | NoisyOrDist | (b, leak, b1, p1, b2, p2, ... bn, pn) | |
| Beta4Dist | (x, a, b, c, d) | | NoisyAndDist | (b, inh, b1, p1, b2, p2, ... bn, pn) | |
| CauchyDist | (x, m, s) | | | | |
| ExtremeValueDist | (x, m, s) | | | | |

Tips


1. Tabel yang dihasilkan oleh persamaan dapat menghasilkan file besar (dan karena itu pembacaannya lambat), jadi Anda mungkin ingin menghapus node's relation table dengan Relation → Remove atau  sebelum menyimpan network ke file. Ketika Anda kemudian membacanya, lakukan Relasi → "Equation to Table" atau  (tanpa node yang dipilih) sebelum menggunakannya.
2. Jika Anda perlu menentukan intermediate variables/ variabel menengah untuk menyederhanakan persamaan, implementasikan sebagai node (intermediate) baru.
3. Jika persamaan menjadi besar, mungkin akan lebih mudah untuk membuatnya dalam editor teks, dan kemudian menempelkannya/ paste ke dalam kotak dialog node.
4. Saat mengedit sebuah persamaan, Anda dapat memotong, menyalin, menempel, membatalkan/ cut, copy, paste, undo dll. menggunakan perintah kunci CTRL [Brent, their???], atau dengan mengklik kanan [???].

3.3 Learning Probability Tables/ Pembelajaran Tabel Probabilitas

Dalam dua bagian sebelumnya, kita belajar bagaimana menentukan hubungan probabilistik antara node dan induknya dengan secara manual mengedit tabel probabilitas atau dengan menulis sebuah persamaan yang merupakan ekspresi singkat untuk tabel semacam itu.

Pada bagian ini kita menemukan cara ketiga agar Netica memungkinkan conditional probabilities ini didefinisikan. Ini dengan mempelajari (*learning*) mereka dari kumpulan dari cases (*collection of cases*). Jika kumpulan cases adalah sampel dari populasi yang kita modelkan, maka kita dapat menggunakan informasi frekuensi yang tersirat dalam data tersebut sebagai perkiraan probabilitas yang diinginkan.

Ini adalah fitur Netica yang sangat kuat dan mudah digunakan. Berikut adalah cara menggunakannya:

1. Kumpulkan data Anda dalam format tabel, satu baris per dataset instance (yaitu, setiap baris mewakili kemunculan dunia yang mungkin).
2. Pastikan nilai data dipisahkan oleh tab atau spasi (pada rilis Netica commas berikutnya juga diterima dan karakter lainnya juga dapat didefinisikan sebagai pembatas).
3. Pastikan data fields yang hilang diganti dengan tanda bintang (*). Pada rilis berikutnya dari Netica, karakter lainnya dapat didefinisikan sebagai karakter data yang hilang.
4. Di bagian atas tabel, buat satu baris dengan nama node. Nama harus berada dalam urutan yang sama dengan kolom data di bagian lain tabel.
5. Akhirnya, di bagian atas case files ada satu baris dengan string "// ~ -> [CASE-1] -> ~". Ini memberitahu Netica bahwa formatnya ada dalam format file Netica Case # 1, sehingga Netica bisa mengetahui tipe file yang sedang dihadapinya. Di masa depan mungkin ada format yang lebih maju.
6. Simpan file yang dihasilkan ke disk dan kemudian load dengan salah satu dari:
 - Relation → "Incorp Case File". (Dalam rilis nest Netica ini adalah Cases → "Incorp Case File".)
 - 

Berikut adalah contoh case file untuk net Asia. Perhatikan bahwa kita telah menambahkan kolom khusus yang disebut 'IDnum'. Hal ini tidak diperlukan, namun merupakan ide bagus untuk penanganan data.

```
// ~->[CASE-1]->~
IDnum  VisitAsia Tuberculosis Smoking Cancer TbOrCa XRay Bronchitis Dyspnea
1      No_Visit Present Smoker Absent True Abnormal Absent Present
2      No_Visit Absent Smoker Absent False Normal Present Present
3      No_Visit * Smoker Present True Abnormal * Present
4      No_Visit Absent NonSmoker Absent False Normal Absent Absent
5      No_Visit Absent Smoker Present True Abnormal Present Present
6      No_Visit Absent Smoker Absent False Abnormal Present Present
...
```

Format CASE files yang tepat diberikan disini [here](#). Ini menggambarkan beberapa fitur bagus lainnya tentang Case files, termasuk bagaimana memberi komentar mereka, menambahkan faktor multiplicity/ keseragaman, dan sebagainya.

Sebagai latihan mengajar, mari kita buat suatu case file untuk mendefinisikan node kita (seperti yang kita lihat bagaimana melakukannya di bagian tutorial B.2.1.6) dan untuk mempelajari probabilitasnya. Lakukan berikut ini:

1. Cut-and-paste teks berikut ke editor teks favorit Anda, mungkin tambahkan beberapa baris Anda sendiri, dan simpan hasilnya sebagai case file Anda.

```
// ~->[CASE-1]->~
Forecast   ActualWeather
rain       rain
cloudy     sunny
cloudy     cloudy
cloudy     rain
sunny      sunny
sunny      sunny
sunny      cloudy
rain       cloudy
```

2. Buat network kosong baru.
3. Pilih Modify → "Add Case File Nodes" (Cases → "Add Case File Nodes ..." di rilis Netica berikutnya), dan berikan case file Anda sebagai masukan. Dua node harus dibangun untuk Anda, *Forecast* dan *ActualWeather*, masing-masing dengan tiga state: hujan, cerah, dan berawan (*rain*, *sunny*, and *cloudy*).
4. Hubungkan Forecast ke ActualWeather (atau ActualWeather ke Forecast: untuk tujuan kita disini tidak ada bedanya; tabel probabilitas akan berakhir dengan menggabungkan korelasi apapun).
5. Pilih Relation → "Incorp Case File" (Cases → "Incorp Case File" dirilis Netica berikutnya), dan sekali lagi pilih file sebagai masukan.
6. Itu semuanya. Lihat tabel yang dihasilkan untuk mengonfirmasi bahwa mereka cocok dengan yang Anda harapkan.

Lebih Lanjut Tentang Learning

Learning Algorithm used. Teori Bayes nets tidak menentukan bagaimana tabel probabilitas dipelajari/ learned. Terdapat banyak *learning algorithms*/ algoritma pembelajaran yang berbeda. Beberapa dikenal sebagai "true Bayesian learning algorithms, Netica menggunakan salah satu dari ini. Ini sederhana dan bekerja dengan baik untuk kebanyakan situasi. Asumsi yang mendasarinya rusak dalam situasi dimana ada banyak induk (dan dengan demikian conditional probability table menjadi besar) dan ada beberapa contoh data. Dalam situasi seperti itu, learning kurang optimal, dan Anda mungkin ingin menemukan cara lain untuk memperkirakan/ menestimasi tabel probabilitas. Untuk rincian tentang algoritma yang tepat untuk digunakan, lihat sistem help/ bantuan dilayar.

Learning from a single case. Cara alternatif untuk menyajikan cases, daripada memuatnya/ loading dari file, adalah menyajikannya satu per satu, kepada Netica sebagai temuan positif/ *positive findings* (*negative* dan *likelihood findings* diabaikan dalam learning semacam ini). Cukup masukkan nilai findings untuk node yang Anda kenal lalu pilih Relation → "Incorporate Case".

Unlearning. Terkadang Anda ingin menghapus cases yang sebelumnya dipelajari. Untuk melakukan ini, pelajari saja seolah-olah itu adalah cases tunggal/ single case (lihat diatas), namun berikan derajat -1. Hasilnya akan sama seperti jika Anda tidak pernah mempelajarinya sejak awal.

Fading/ Forgetting. Terkadang Anda ingin bias/ berprasangka apa yang Anda pelajari menuju cases terbaru. Dengan Netica, Anda bisa memilih untuk "*fade*/ memudahkan" pengetahuan lama secara bertahap. Untuk melakukan ini, pilih Relation → Fade, dan masukkan sebuah tingkat/ derajat, d, dari 0 ke 1. Netica kemudian akan mengurangi pengalaman dan memperlancar probabilitas node yang dipilih dengan jumlah yang ditentukan oleh derajat itu. 0 tidak berpengaruh, dan 1 menciptakan distribusi seragam tanpa pengalaman, sehingga bisa melepaskan semua pembelajaran/ learning sebelumnya. Fading dua kali, dengan d1 dan kemudian d2, sama dengan memudar sekali oleh $d1 * d2$. Lihat help di layar untuk informasi lebih lanjut.

C. Intermediate Topics/ Topik Menengah

1. Membangkitkan Cases Acak/ Random Cases

2. Pengambilan Keputusan Berurutan/ Sequential Decision Making

3. Sensitivitas terhadap Temuan/ Sensitivity to Findings

Pertanyaan Terbaik Selanjutnya

4. Penamaan Link/ Naming Links

Untuk Generalisasi Persamaan


Untuk Membuat Net-snippet Libraries

Dalam Pengembangan :

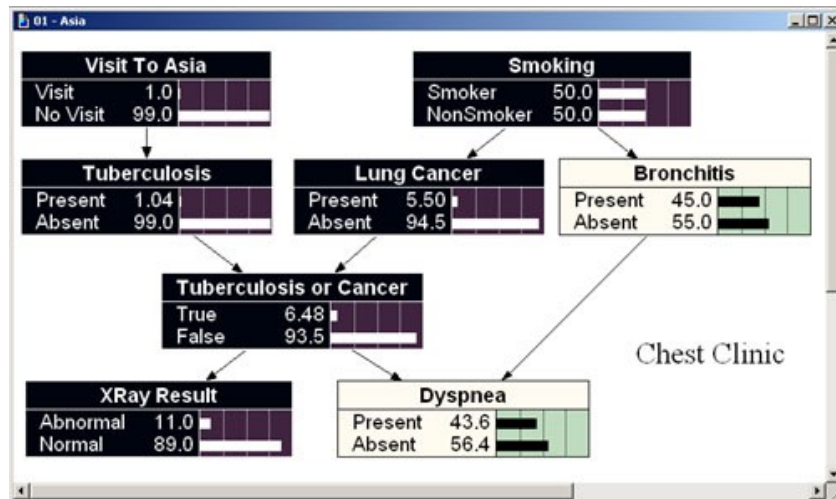
5. Membangun Good Model/ Desain Net yang Baik
6. Membangkitkan Presentation Quality Graphics
7. Isu Ukuran - Model Besar
8. Probabilitas Bersyarat Node/ Node Conditional Probabilities
9. Laporan

1. Membangkitkan Random Cases

Pada bagian B.3 kita melihat bagaimana Netica dapat mempelajari *conditional probabilities* antara node dalam model dunia dengan membaca dalam sekumpulan nilai sampel yang diambil secara acak dari dunia itu. Dalam tutorial ini kita tunjukkan bagaimana Anda juga bisa melakukan yang sebaliknya. Mengambil yang diberikan Bayes nets, dan buatlah serangkaian nilai sampel yang sesuai dengan distribusi probabilitas yang sama yang ditemukan di net. Proses ini dikenal sebagai *simulation*, atau kadang sebagai *sampling*. Dengan Netica, menggambar contoh simulasi dari dunia model sangat mudah. Sebenarnya, hanya beberapa klik tombol yang diperlukan. Prosesnya akan create/ membuat sebuah file dari *sample cases*. File ini kemudian bisa digunakan untuk *learning* atau untuk aplikasi lain. Berikut adalah prosedur untuk membangkitkan file

1. Kompilasi net (klik .
2. Pilih semua node yang Anda inginkan untuk memiliki nilai dalam *case file*. Untuk melakukan ini, klik pada masing-masing node, sambil menahan tombol Ctrl di keyboard Anda, atau tarik persegi panjang di sekitar node yang diinginkan (atau lakukan kombinasi kedua operasi ini). Perhatikan bahwa semua node dari network akan digunakan untuk menghasilkan case, namun kolom hanya akan dibuat (output) untuk yang dipilih.
3. Klik Network → Simulate-Cases (Catatan: pada rilis Netica berikutnya, fungsi ini akan berpindah dari menu Network ke menu Cases).
4. Dialog akan dimunculkan, meminta Anda untuk memasukkan jumlah cases yang ingin Anda hasilkan. Masukkan angka lebih besar dari nol.
5. Dialog kedua akan muncul, dialog file "Save As" Windows standar, meminta Anda untuk memilih di direktori mana dan dengan nama apa untuk membuat file. Pilih ini dan klik "Save".
6. Akhirnya, dialog ketiga akan dimunculkan, menanyakan berapa persentase entri yang ingin Anda lewatkan data/ missing data. Biasanya Anda akan memasukkan 0 untuk jumlah data yang hilang, namun jika Anda ingin memiliki cases file dengan tanda bintang untuk beberapa fraksi/ pecahan dari kolom, masukkan fraksi tersebut sebagai persentase (25 => satu kuartal/ bagian).
7. File kemudian akan ditulis kedalam disk, seperti yang diinginkan. Ini adalah file teks standar (ascii), yang dapat Anda baca dengan editor teks apapun, atau Anda dapat menekan F8 untuk melihat-lihat individual cases dan melihat seperti apa bentuknya jika masuk kedalam Net sebagai evidence.

Mari kita coba ini. Kita akan menggunakan Asia lagi, karena kita sudah mengenalnya. Tapi prosedurnya identik untuk setiap Bayes nets. Katakanlah kita memilih semua node didalam Asia, kecuali Dyspnea dan Bronchitis, dan kita meminta 100 cases dihasilkan dengan 25% sebagai data yang hilang.



File yang dihasilkan akan terlihat seperti ini (cases file yang Anda dapatkan kemungkinan besar akan berbeda, karena random numbers terlibat; juga, format kolom mungkin sedikit berbeda, karena perbedaan spesifikasi tab pada platform yang berbeda):

```
// ~->[CASE-1]->~
```

```
// File created by an unlicensed user using Netica 1.12
// on Oct 21, 2002 at 10:36:22.
```

| IDnum | Tuberculosis | TbOrCa | Cancer | VisitAsia | Smoking | XRay |
|-------|--------------|--------|--------|-----------|-----------|--------|
| 1 | Absent | False | Absent | No_Visit | Smoker | Normal |
| 2 | * | False | Absent | No_Visit | * | Normal |
| 3 | Absent | * | Absent | No_Visit | NonSmoker | * |
| 4 | Absent | False | * | No_Visit | NonSmoker | Normal |
| 5 | Absent | False | Absent | * | Smoker | * |
| 6 | Absent | * | Absent | * | Smoker | * |
| 7 | * | False | * | No_Visit | NonSmoker | * |
| 8 | Absent | False | Absent | No_Visit | * | Normal |
| 9 | * | False | Absent | No_Visit | NonSmoker | Normal |
| 10 | Absent | False | * | No_Visit | Smoker | Normal |
| ... | | | | | | |
| 98 | Absent | * | * | No_Visit | * | Normal |
| 99 | Absent | False | Absent | No_Visit | Smoker | * |
| 100 | Absent | False | Absent | No_Visit | Smoker | Normal |

Catatan Tentang Simulasi

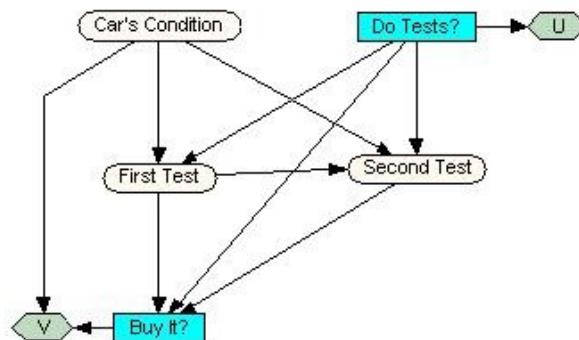
- Algoritma sampling yang digunakan tepat, sehingga frekuensi jarak jauh dari cases akan mendekati probabilitas belief network dengan tepat, dengan mempertimbangkan semua temuan/ findings yang saat ini masuk.
- **Dengan Persamaan:** Jika satu atau lebih node memiliki persamaan untuk menentukan hubungan antara node dan induknya, maka Anda mungkin ingin Netica menggunakan persamaan tersebut secara langsung untuk menghasilkan random cases, bukannya tabel probabilitas yang mendekati/ mrngirakan persamaan. Dalam hal ini, jangan compile network sebelum melakukan Network → Simulate-Cases.
- Proses pengambilan sampel akan lambat jika network memiliki seperangkat temuan/ findings yang tidak mungkin masuk (metode penolakan/ rejection digunakan).
- Dalam case file yang dihasilkan/ dibangkitkan, variabel continuous (terlepas dari apakah mereka telah atau tidak didiskritisasi/ discretized) akan memiliki nilai bilangan real continuous mereka untuk setiap cases, bukan hanya suatu state/ keadaan yang mewakili serangkaian nilai.

2. Pengambilan Keputusan Berurutan/ Sequential Decision Making

Pada bagian 1.4 kita melihat sebuah keputusan net yang sangat sederhana, Umbrella. Itu hanya satu *decision node*, dan hanya satu *utility node*. Dalam tutorial ini kita akan melihat kumpulan keputusan yang lebih kompleks, yang menggambarkan *sequential decisions* dan *multiple utility nodes*.

"Car Buyer" adalah decision nets yang akan membantu kita membeli mobil. Ini akan membantu kita dalam menentukan terlebih dahulu apakah mobil tersebut menjamin pengujian/ car warrants testing, sebelum memutuskan untuk membeli, dan jika tes tersebut dilakukan, apakah akan membeli atau tidak. Perhatikan bahwa ada dua *decision node* dan dua *utility node*.

Car Buyer - Neapolitan



Ini dari Neapolitan90 (hal.373), yang merupakan versi sederhana dari contoh pembeli mobil dari Howard62.

Menghilangkan alasan yang diberikan disana bagaimana angka-angka tersebut sampai, kita memiliki cerita berikut:

Joe akan membeli mobil bekas, yang bisa bagus dengan probabilitas 0.8 atau tidak memuaskan dengan probabilitas 0.2. Setelah memperhitungkan perbaikan, keuntungan Joe akan menjadi \$ 60 jika mobil itu bagus, dan \$ -100 jika itu buruk. Sebelum membeli mobil ia memiliki pilihan untuk memiliki satu tes atau dua tes yang dilakukan di atasnya. Biaya tes pertama \$ 9, dan keduanya sama-sama berharga \$ 13. Tes pertama memiliki kesempatan 90% untuk kembali positif jika mobil itu bagus, dan kemungkinan 40% jika itu adalah tidak memuaskan. Joe memiliki 2 keputusan untuk dibuat: apakah akan melakukan tes, dan apakah akan membeli mobil. Ini ditunjukkan oleh "Do Testing?" dan "Buy It?" decision nodes. Hasil tes diberikan oleh node "First Test" dan "Second Test". Biaya tes diwakili oleh utility node U, dan keuntungan setelah perbaikan (tidak termasuk biaya uji) oleh utility node V.

Sekarang mari kita meninjau kembali setiap detail ini pada gilirannya dan melihat bagaimana kita mengumpulkan/ memasang network kita dengan tepat.

1. Joe akan membeli mobil bekas, yang bisa bagus dengan probabilitas 0.8 atau tidak memuaskan dengan probabilitas 0.2:

| Good | Lemon |
|--------|--------|
| 80.000 | 20.000 |

2. Setelah memperhitungkan perbaikan, keuntungan Joe akan menjadi \$ 60 jika mobil itu bagus, dan \$ -100 jika itu buruk:

Table (in net Car_Buyer_Neapolitan)

Node: V

Deterministic

| C | B | V |
|-------|----------|------|
| Good | Buy | 60 |
| Good | Dont_Buy | 0 |
| Lemon | Buy | -100 |
| Lemon | Dont_Buy | 0 |

3. Sebelum membeli mobil ia memiliki pilihan untuk memiliki satu tes atau dua tes yang dilakukan atasnya. Biaya tes pertama \$ 9, dan keduanya sama-sama berharga \$ 13:

U Table (in net Car_Buyer_Neapolitan)

Node: U

Deterministic

| D | U |
|-------|-----|
| None | 0 |
| First | -9 |
| Both | -13 |

4. Tes pertama memiliki kesempatan 90% untuk kembali positif jika mobil itu bagus, dan kemungkinan 40% jika itu adalah tidak memuaskan:

FirstTest Table (in net Car_Buyer_Neapolitan)

Node: FirstTest

Chance

| C | D | Not_Done | Positive | Negative |
|-------|-------|----------|----------|----------|
| Good | None | 100.00 | 0.000 | 0.000 |
| Good | First | 0.000 | 90.000 | 10.000 |
| Good | Both | 0.000 | 90.000 | 10.000 |
| Lemon | None | 100.00 | 0.000 | 0.000 |
| Lemon | First | 0.000 | 40.000 | 60.000 |
| Lemon | Both | 0.000 | 40.000 | 60.000 |

5. Jika tes pertama hasil positif, maka tes kedua memiliki kemungkinan 88,89% untuk mengembalikan positif jika mobil itu bagus, dan peluang 33,33% jika itu adalah tida memuaskan. Jika tes pertama hasil negatif, maka tes kedua memiliki kesempatan 100% untuk mengembalikan positif jika mobil itu bagus, dan 44,44% kemungkinan jika itu adalah tida memuaskan:

SecondTest Table (in net Car_Buyer_Neapolitan)

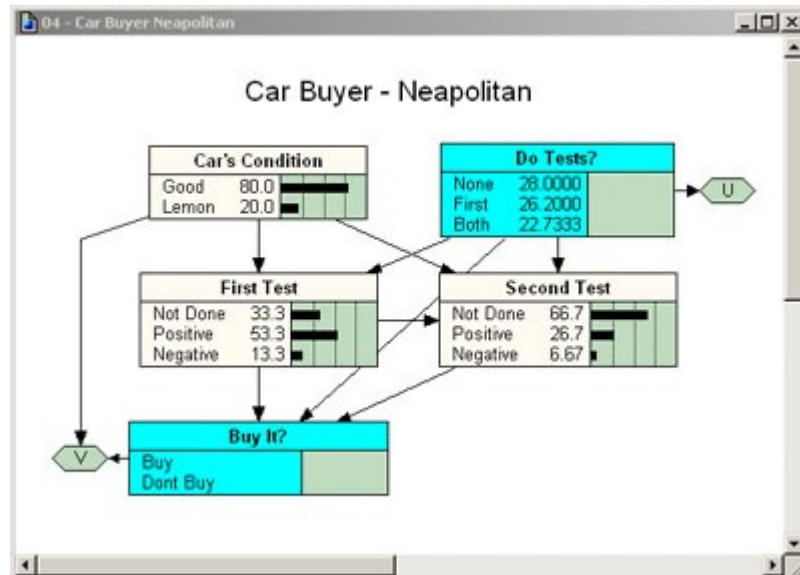
Node: SecondTest

Chance

| T | CarCond... | D | Not_Done | Positive | Negative |
|----------|------------|-------|----------|----------|----------|
| Not_Done | Good | None | 100.00 | 0.000 | 0.000 |
| Not_Done | Good | First | x | x | x |
| Not_Done | Good | Both | x | x | x |
| Not_Done | Lemon | None | 100.00 | 0.000 | 0.000 |
| Not_Done | Lemon | First | x | x | x |
| Not_Done | Lemon | Both | x | x | x |
| Positive | Good | None | x | x | x |
| Positive | Good | First | 100.00 | 0.000 | 0.000 |
| Positive | Good | Both | 0.000 | 88.889 | 11.111 |
| Positive | Lemon | None | x | x | x |
| Positive | Lemon | First | 100.00 | 0.000 | 0.000 |
| Positive | Lemon | Both | 0.000 | 33.333 | 66.667 |
| Negative | Good | None | x | x | x |
| Negative | Good | First | 100.00 | 0.000 | 0.000 |

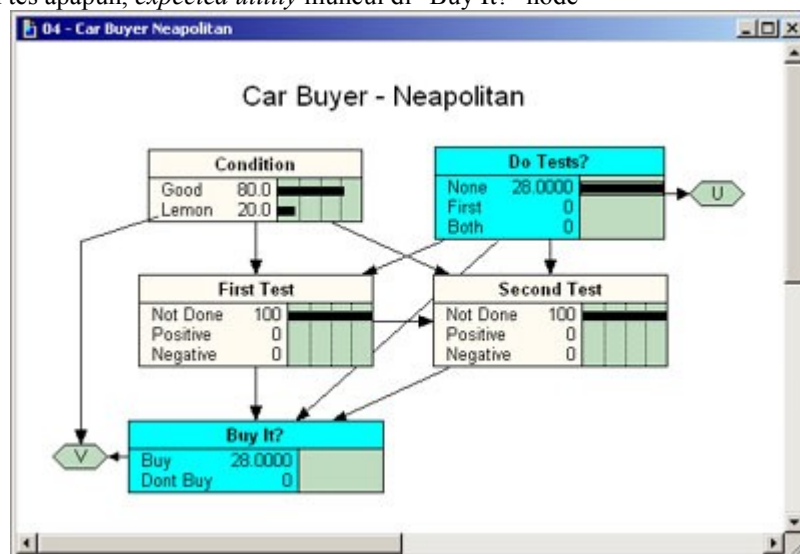
Ketika Joe memutuskan untuk melakukan tes, dia tidak tahu nilai dari variabel-variabel ini, jadi tidak ada link yang masuk ke "Do Testing?" node. Ketika dia memutuskan apakah akan membeli, dia akan mengetahui hasil dari kedua tes tersebut (hasilnya mungkin "not done/ tidak dilakukan"), dan karena itu ada link dari dua node tersebut menjadi "Buy It?". Dia juga akan tahu nilai "Do Testing?" karena dia sudah membuat keputusan itu, jadi Anda bisa meletakkan link dari node itu ke "Buy It?", tapi ini tidak diperlukan karena ini adalah forgetting link/ link yang tidak terlupakan dan sudah ada jalur yang diarahkan dari "Do Testing?" untuk "Buy It?".

Sekarang mari kita kompilasi network. Karena auto-update dihidupkan, maka akan segera dipecahkan. Hasilnya adalah:

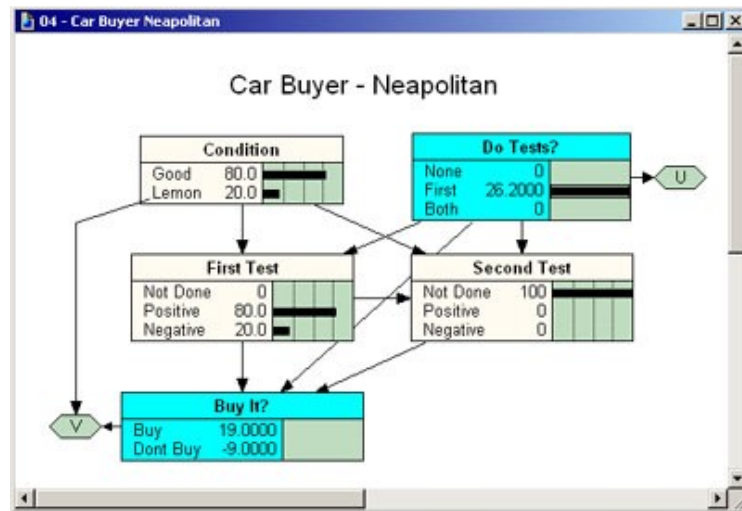


Perhatikan bahwa Netica menambahkan link no-forgetting dari "Do Testing?" untuk "Buy It?", menunjukkan bahwa "Do Tests?" mungkin relevan dengan keputusan kedua, hanya berdasarkan struktur link lainnya (tapi ternyata tidak demikian). Utilitas yang diharapkan/ *expected utility* dari setiap pilihan keputusan untuk "Do Tests?" dicetak didalam node; Tidak melakukan tes apa pun memiliki utilitas yang diharapkan/ *expected utility* tertinggi dari 28, jadi ini adalah pilihan terbaik. Tidak ada *expected utility* yang tercetak di "Buy It?" node, sejak keputusan pertama belum dibuat.

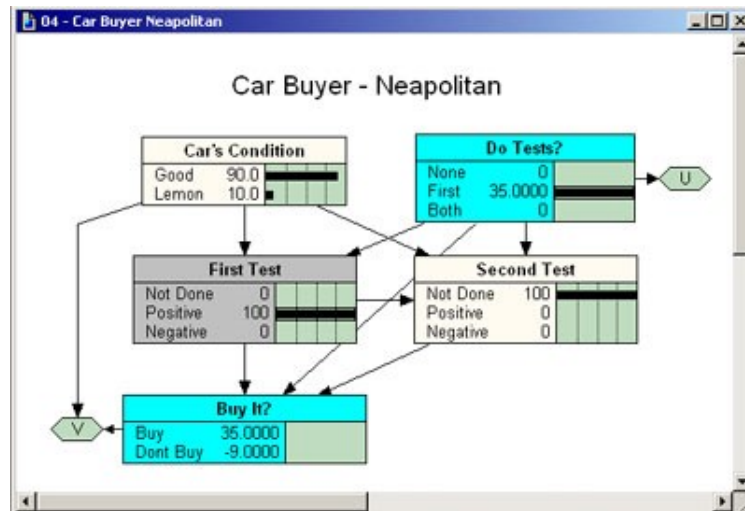
Jika Anda klik pada keputusan "None" dari "Do Tests?" untuk menunjukkan bahwa Joe memutuskan untuk tidak melakukan tes apapun, *expected utility* muncul di "Buy It?" node



Pilihan terbaik adalah membelinya, menghasilkan *expected utility* tertinggi - keuntungan keseluruhan sebesar \$ 28. Coba klik "First" dari "Do Tests?" node, menunjukkan bahwa Joe memutuskan untuk hanya mendapatkan tes pertama yang dilakukan.

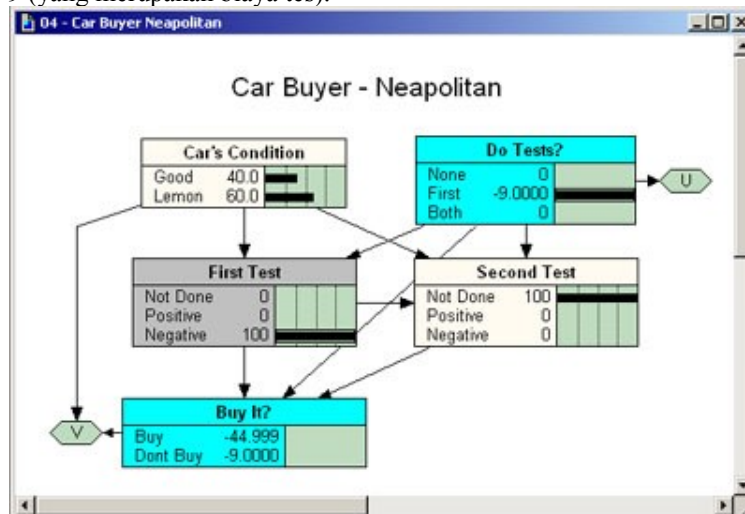


Expected utility secara keseluruhan adalah 26,2. Kemudian klik menunjukkan bahwa hasil tes pertama adalah "Positive".



Keputusan terbaik adalah membeli dan utilitas yang diharapkan telah meningkat menjadi 35.

Akhirnya, jika tes pertama keluar negatif, maka keputusan terbaik adalah tidak membeli, dan utilitas yang diharapkan adalah -9 (yang merupakan biaya tes).

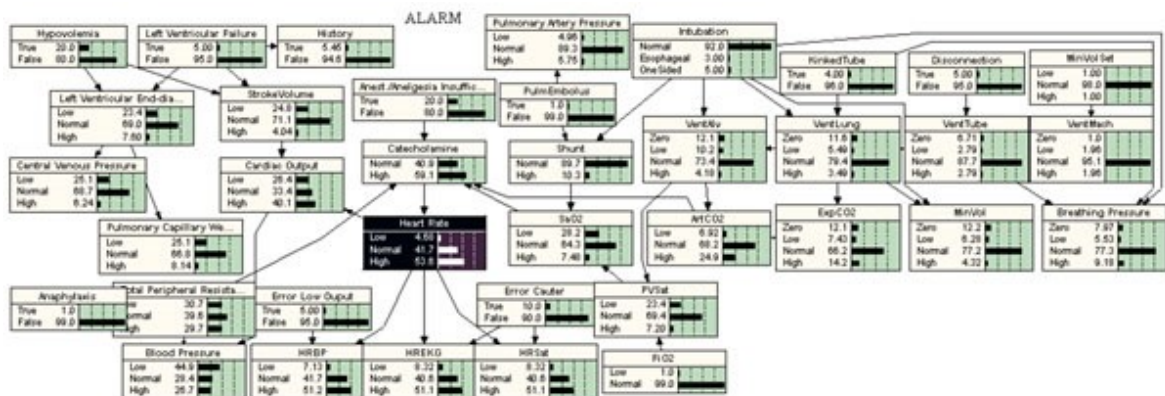


Cobalah bereksperimen dengan kombinasi dari findings dan pilihan keputusan lainnya.

3. Sensitivitas Terhadap Temuan/ Sensitivity to Findings

Terkadang berguna untuk mengetahui seberapa besar kepercayaan kita/ our belief pada node tertentu dipengaruhi oleh temuan/ findings pada node lain. Kita ingin tahu seberapa sensitif kepercayaan kita terhadap nilai node ini terhadap temuan node lain. Jika sangat sensitif, mungkin kita inginkan dan penting bagi kita untuk mengetahui keadaan node itu, maka kita mungkin ingin menginvestasikan usaha/ effort dalam menentukan nilai dari semua node yang secara substansial mempengaruhinya. Netica dapat menghitung sensitivitas node terhadap temuan dengan mudah.

Dalam tutorial ini kita akan melihat net ALARM. 'ALARM' adalah singkatan dari 'A Logical Alarm Reduction Mechanism/ Mekanisme Pengurang Alarm Logis'. Ini adalah sistem pesan alarm diagnostik medis untuk pemantauan pasien dan dijelaskan didalam BeinlichSCC89. Ini adalah aplikasi Bayes nets dunia nyata. Terlalu rumit untuk dijelaskan disini, tapi itu sudah cukup untuk tujuan kita. Anda mungkin ingin membawanya kedalam Netica untuk diajak kerja sama, untuk visibilitas yang lebih baik.



Misalkan kita ingin mengetahui node mana yang paling mempengaruhi pengetahuan kita tentang "Heart Rate". Klik di bagian atas area judul dari kotak "Heart Rate" untuk memilihnya, lalu pilih Network → Sensitivity to Findings.

Laporan terperinci akan diproduksi di jendela "Netica - Messages" (jika Anda tidak dapat menemukannya, klik Window → Messages untuk memunculkannya). Inilah laporannya:

Probability of new evidence = 100 %, of all evidence = 100 %.

Sensitivity of 'HR' to findings at 'HR':

| Probability ranges: | Min | Current | Max | RMS Change |
|---------------------|-----|---------|-----|------------|
| Low | 0 | 0.04678 | 1 | 0.2112 |
| Normal | 0 | 0.4169 | 1 | 0.493 |
| High | 0 | 0.5363 | 1 | 0.4987 |

Quadratic scoring = 0.2996

Entropy reduction = 1.215 (100 %)

Sensitivity of 'HR' to findings at 'HRBP':

| Probability ranges: | Min | Current | Max | RMS Change |
|---------------------|-----------|---------|--------|------------|
| Low | 0.0009137 | 0.04678 | 0.6431 | 0.1652 |
| Normal | 0.008143 | 0.4169 | 0.9609 | 0.4617 |
| High | 0.03796 | 0.5363 | 0.9909 | 0.4671 |

Quadratic scoring = 0.2467

Entropy reduction = 0.9799 (80.7 %)

[Singkatnya, sebagian besar bagian rincian laporan tidak ditunjukkan di sini]

Sensitivity of 'HR' to findings at 'PAP':

| Probability ranges: | Min | Current | Max | RMS Change |
|---------------------|---------|---------|---------|------------|
| Low | 0.04307 | 0.04678 | 0.04701 | 0.0009153 |
| Normal | 0.384 | 0.4169 | 0.419 | 0.008136 |
| High | 0.534 | 0.5363 | 0.573 | 0.009052 |

Quadratic scoring = 7.235e-005

Entropy reduction = 0.0002387 (0.0196 %)

Sensitivity of 'HR' due to a finding at another node:

| Node | Mutual Info | Quadratic Score |
|--------------|-------------|-----------------|
| HR | 1.21495 | 0.2995618 |
| HRBP | 0.97993 | 0.2466639 |
| HREKG | 0.81844 | 0.2124077 |
| HRSat | 0.81844 | 0.2124077 |
| Catechol | 0.68583 | 0.1587396 |
| CO | 0.52149 | 0.1401417 |
| SaO2 | 0.16188 | 0.0461771 |
| PVSat | 0.12620 | 0.0362042 |
| VentAlv | 0.11472 | 0.0330883 |
| ArtCO2 | 0.11419 | 0.0333368 |
| ExpCO2 | 0.10006 | 0.0295365 |
| InsuffAnesth | 0.09786 | 0.0283655 |
| BP | 0.09734 | 0.0268312 |
| TPR | 0.09482 | 0.0282133 |
| VentLung | 0.07704 | 0.0224051 |
| MinVol | 0.06422 | 0.0189986 |
| Press | 0.04806 | 0.0143849 |
| Shunt | 0.04013 | 0.0117534 |
| VentTube | 0.03752 | 0.0109797 |
| Intubation | 0.03224 | 0.0093943 |
| Disconnect | 0.01694 | 0.0049907 |
| KinkedTube | 0.01427 | 0.0041830 |
| VentMach | 0.01091 | 0.0032041 |
| MinVolSet | 0.00360 | 0.0010587 |
| FiO2 | 0.00304 | 0.0009002 |
| PulmEmbolus | 0.00254 | 0.0007571 |
| Anaphylaxis | 0.00208 | 0.0006266 |
| PAP | 0.00024 | 0.0000723 |
| LVEDVolume | 0.00000 | 0.0000000 |
| StrokeVolume | 0.00000 | 0.0000000 |
| LVFailure | 0.00000 | 0.0000000 |
| Hypovolemia | 0.00000 | 0.0000000 |
| ErrLowOutput | 0.00000 | 0.0000000 |
| CVP | 0.00000 | 0.0000000 |
| PCWP | 0.00000 | 0.0000000 |
| ErrCauter | 0.00000 | 0.0000000 |
| History | 0.00000 | 0.0000000 |

Arti sebenarnya dari semua hal dalam laporan ini dijelaskan dalam bantuan/ help layar Netica.

Pada akhir laporan adalah daftar semua node, dengan temuan yang paling mungkin menghasilkan perubahan terbesar dalam kepercayaan dari Heart Rate yang tercantum pertama. Kolom angka pertama memberikan tingkat numerik.

Perhatikan, bahwa tidak disangka-sangka, nodes "Heart Rate" yang paling berpengaruh, adalah induk dan anaknya.

Catatan Tentang Sensitivitas Terhadap Temuan

- Laporan tersebut mengambil temuan/ findings yang saat ini masuk dalam akun.
- Node disebut dengan nama, bukan dengan judul/ title. Ini karena nama berukuran terbatas dan lebih mudah untuk format. Anda dapat menampilkan node dengan nama mereka daripada title dengan memilih Style → Name.

4. Penamaan Links

4.1 Untuk Generalized Equations

Saat pertama kali memulai bekerja dengan persamaan, Anda mungkin akan menggunakan nama dari node induk dalam persamaan Anda. Namun, terkadang Anda menginginkan representasi lokal yang lebih banyak, sehingga Anda bisa melepaskan/ disconnect beberapa induk dan menghubungkan nodenya dengan induk baru tanpa harus mengubah semua nama node dalam persamaan. Atau Anda mungkin ingin menyalin dan menempelkan/ copy and paste persamaan dari satu node ke node yang lain, tanpa mengubah semua nama node.

4.2 Untuk Membuat Library Net-snippet/ guntingan/ potongan

Atau mungkin Anda banyak ingin copy and paste node untuk digunakan dengan induk baru. Atau untuk menempatkan node di network library tanpa induk, sehingga akan dipasok dengan induk baru saat digunakan.

Solusi untuk semua masalah ini adalah dengan menggunakan *link names*, kadang disebut *input names*. Mereka memberikan nama argumen untuk setiap link yang memasuki node (dan oleh karena itu merupakan proxy untuk setiap node induk). Anda dapat mengaturnya dengan kotak dialog node. Cukup tentukan *Multi-Purpose box* menjadi "*Input Name*" dan berikan nama, satu per baris, atau koma dan / atau spasi yang terpisah (nama linknya adalah IDNames, jadi dibatasi hingga 30 karakter atau kurang dan tidak boleh berisi spasi atau tanda baca lainnya, dan harus dimulai dengan karakter alfabet). Anda merujuk mereka dalam persamaan Anda dengan cara yang persis sama dengan nama induk Anda. Bila induk terputus/ disconnected, nama link akan tetap ada.

Catatan. Jika nama link didefinisikan untuk sebuah node, maka nama tersebut harus digunakan sebagai pengganti nama induknya.

Sebuah node dapat memiliki beberapa link yang dinamai, dan yang lainnya tidak disebutkan namanya.

Exercise. Cobalah memberi nama pada beberapa link dari salah satu test node Anda. Kemudian putuskan/ disconnect node itu (dengan menghapus node yang memiliki link ke dalamnya). Perhatikan bagaimana nama link yang diawetkan/ dipertahankan. Kemudian potong/ cut dan tempelkan/ paste potongan net itu ke tempat lain di network, atau ke network lain dan buat nodes baru untuk dilampirkan/ attach pada cuplikan/ snippet yang disalin. Sekali lagi, perhatikan bagaimana snippet menyimpan nama linknya.

Membuat net-snippet libraries adalah penggunaan Netica yang kuat (cuplikan net/ net-snippets hanyalah network reguler, namun ditujukan untuk penggunaan ulang/ re-use), dan penamaan link membuat mereka lebih mudah untuk mengungkapkan secara konseptual dan lebih sederhana untuk bekerja dengannya. Dengan setiap Net yang Anda buat, Anda mungkin ingin memikirkan segmentasi cuplikan yang dapat digunakan didalam aplikasi Bayes net lainnya. Ini berlaku, sub-worlds didalam dunia.

D. Advanced Topics/ Topik Lanjutan

1. Data Hilang dan Variabel Tersembunyi/ Missing Data and Hidden Variables

EM Learning

Gradient Descent Learning, Neural Nets

2. Menguji Nets Dengan Cases

3. Mengelola Case Case files

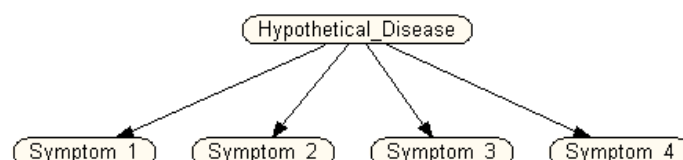
Dalam Pengembangan :

4. Arah Link/ Link Direction
 - Masalah klasifikasi
 - Pemodelan kausal, independensi dan modularitas
5. Ekspansi Waktu
6. Pola Desain Net Khusus
 - Diagnostic Nets
 - Decision Nets
7. Node Absorption/ Penyerapan Node
8. Junction Trees (penghapusan ordering)
9. Nilai Informasi
 - Berapa biaya yang harus diperoleh informasi lebih lanjut?
10. Case-Based Reasoning
 - Menggunakan Bayes nets untuk membantu penalaran berbasis cases
11. Constraint Satisfaction
 - Menggunakan Bayes nets untuk melakukan constraint satisfaction
12. Logika fuzzy
 - Kelemahan logika fuzzy
 - "Berpikir adalah Bahasa Alami/ Natural Language " illusion
 - "Tinggi/ Tall" bukan suatu fuzzy set, tapi indikator bahwa indikator lebih tinggi daripada yang mungkin Anda pikirkan
13. Menghubungkan Bayes nets dengan Netica API

1. Data Hilang dan Variabel Tersembunyi/ Missing Data and Hidden Variables

EM dan Gradient Descent learning adalah dua teknik untuk mempelajari variabel "tersembunyi/ *hidden*" (juga dikenal sebagai "*latent*"), yaitu variabel yang tidak Anda observasi, namun Anda menduga ada dan berguna untuk memodelkan dunia Anda.

Contoh situasi seperti ini adalah ketika Anda menduga bahwa sejumlah nodes mungkin memiliki penyebab umum yang tidak dapat diamati, seperti penyakit yang tidak dapat diamati secara langsung, sehingga menimbulkan sejumlah gejala, seperti pada:



Dalam cases ini, kita memiliki sejumlah gejala yang kita amati pada pasien kita, dan yang tampaknya berkorelasi. Jadi kita menebaknya mungkin ada penyakit yang bisa menjelaskan gejalanya. Tidak semua pasien dengan penyakitnya tentu saja akan mengekspresikan semua gejalanya. Kita pada dasarnya menebak bahwa ada penyakit yang mendasarinya. Sebenarnya ada lebih dari satu penyakit, atau beberapa alasan lain yang bisa menjelaskan korelasi (dokter jahat bisa menyuntikkan pasiennya dengan obat-obatan yang efek sampingnya meniru penyakit).

How many states to choose for the hidden variable?/ Berapa banyak state yang memilih variabel tersembunyi?. Ketika Anda membangun net dengan Netica, Anda menambahkan node untuk variabel latent, dan berikan sejumlah state yang Anda anggap seharusnya ada. Anda mungkin harus mengulangi keseluruhan proses ini, dengan jumlah state yang berbeda setiap kali Anda menjalankan *learning*, untuk melihat apa yang paling sesuai dengan data. Lebih baik berbuat salah karena terlalu sedikit state, karena terlalu banyak state akan sulit bagi Netica untuk belajar/ learn, dan agar Anda bisa menafsirkannya. Ingatlah bahwa node Netica belajar/ learn mungkin akan memiliki state didalam urutan yang berbeda dari yang Anda harapkan, karena Netica tidak memiliki pengetahuan yang akan membantunya memutuskan sebuah perintah untuk state.

Anda mungkin bisa menebak dari paragraf diatas bahwa mempelajari/ learning properti optimal dari variabel tersembunyi secara hipotetis/ *hypothetical hidden variable* bukanlah masalah sederhana untuk menekan satu tombol dan membiarkan Netica melakukan semua pekerjaan untuk Anda. Netica akan membantu, tapi Anda perlu membantu Netica dengan mengarahkan solusi terbaik secara keseluruhan.

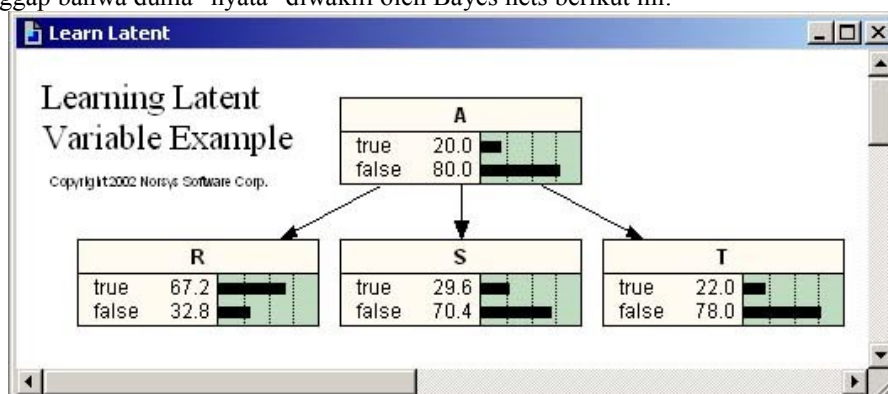
Note #1. EM dan Gradient Descent learning hanya tersedia dengan versi 2.17 dan Netica yang lebih baru.

Note #2. Jika tidak ada variabel tersembunyi di network Anda, atau sejumlah besar data yang hilang dalam case files Anda, Anda harus melakukannya **not/** tidak menggunakan EM atau Gradient Descent learning. Hasilnya mungkin serupa dengan regular learning/ pembelajaran reguler, dan akan memakan waktu lebih lama untuk dihitung.

1.1 EM Learning

Dalam latihan ini, kita mendemonstrasikan bagaimana mempelajari variabel *latent* (atau "*hidden/ tersembunyi*") dengan menggunakan EM learning.

Mari kita anggap bahwa dunia "nyata" diwakili oleh Bayes nets berikut ini:



Dengan ini kita maksudkan, bahwa dunia nyata yang ingin kita modelkan diatur oleh conditional probability distributions yang melekat pada network di atas. Download network ini [here](#) dan load kedalam Netica.

Berikut ini kita akan mencoba mempelajari network kedua, Net #2, menggunakan EM learning, dan melihat apakah network yang dihasilkan mengandung kemiripan dengan network asli diatas. Jika demikian, kita kemudian dapat mengatakan bahwa kita telah melakukan pekerjaan yang baik dalam mempelajari/ learning struktur dunia "nyata".

Case file berikut mematuhi statistik dunia "nyata" kita. Perhatikan bahwa missing/ tidak ada referensi ke A, yang merupakan node yang tidak dapat diamati di dunia nyata kita.

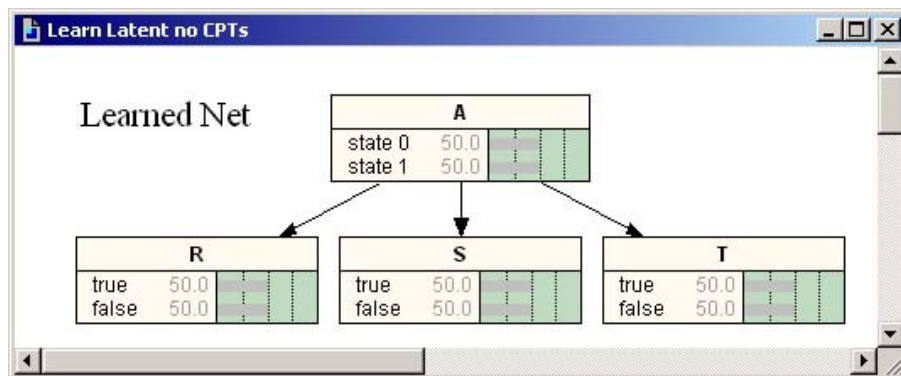
```
// 100K cases following the distribution of the "Learn_Latent.dne" Bayes net
NumCases      R      S      T
22912         false  false  false
4096          false  false  true
4688          false  true   false
1104          false  true   true
35808         true   false  false
7582          true   false  true
2             true   false  *
14592         true   true   false
9216          true   true   true
```


Tugas EM learning adalah menggunakan cases file di atas yang dapat diamati untuk melihat apakah mungkin ada variabel keempat, katakan A, dapat menjelaskan R, S, dan T.

Anda dapat cut and paste teks di atas ke dalam file, Learn_Latent_Data.cas. Atau, jika Anda mau, Anda bisa menghasilkan yang serupa dari network di atas, dengan memilih node R, S, T, dan kemudian memilih "Cases → Simulate Cases". Kemudian save hasilnya seperti file Learn_Latent_Data.cas. File seperti itu akan memiliki 100.000 garis di dalamnya, namun frekuensi dari delapan cases yang mungkin harus serupa secara statistik seperti di atas.

Langkah selanjutnya adalah membangun struktur dasar dari 2nd net kita.

Dengan Netica, kita membangun sebuah network kosong baru, dan memberikan empat node, tiga untuk variabel dalam database kita yang dapat diamati, dan satu sebagai penyebab hipotetis kita untuk tiga lainnya:



Jadi, untuk meninjau kembali apa yang kita lakukan: dugaan kita adalah bahwa ada penyebab umum, A, untuk mengobservasi R, S dan T, dan kita ingin menentukan bagaimana R, S dan T bergantung secara probabilistik pada A.

Anda bisa membangun network baru diatas dari awal, atau hanya menghapus CPT dari network "Learn_Latent", jika Anda ingin menghemat waktu.

Kita sekarang siap untuk melakukan beberapa pembelajaran EM.

Pilih "Cases → Learn Using EM" dan pilih file "Learn Latent Data.cas" dari kotak dialog, dan setuju default degree dari 1. Netica akan membuka jendela Messages dan melanjutkan untuk melakukan banyak langkah algoritma EM untuk mempelajari/ learn CPT baru. Untuk setiap langkah dari EM, ia mencetak garis yang menunjukkan nomor iterasi, "kemungkinan log/ log likelihood", dan persentase perubahan log likelihood dari iterasi sebelumnya. "log likelihood" adalah rata-rata per case negatif dari logaritma dari probabilitas cases diberikan Bayes net saat ini (struktur + CPT).

Biasanya Anda hanya akan melakukan beberapa langkah algoritma EM, dan kemudian menghentikan proses dengan menekan tombol kiri mouse dan tombol <ctrl> dan <alt> untuk beberapa saat. Namun, dalam contoh kita, program berjalan sampai selesai cukup cepat.

Anda sekarang dapat membandingkan learned net dengan yang asli (yang mewakili distribusi di dunia nyata). Sebagai contoh, pilih node R dari network asli dan pilih Table → View/ Edit, kemudian lakukan hal yang sama untuk node R dari network Learned. Anda dapat melihat bahwa learned net telah mempelajari hubungan dunia nyata/ real-world relationships dengan baik mengingat bahwa ia tidak memiliki observasi terhadap node A.

Catatan untuk Siswa dari Learning Theory:

Algoritma EM mencari CPT Bayes net dalam usaha untuk memaksimalkan probabilitas data yang diberikan pada Bayes net (yaitu meminimalkan kemungkinan log negatif). Ini juga bisa dilakukan dengan menggunakan algoritma gradient descent, dijelaskan selanjutnya.

1.2 Gradient Descent Learning, Neural Nets

Descent Gradient bekerja mirip dengan EM Learning hanya menggunakan algoritma yang sangat berbeda secara internal (yang mirip dengan propagasi balik Neural Net/ Neural Net back propagation). Setelah Anda mengeksplorasi EM learning, Anda mungkin ingin mencoba Gradient Descent learning untuk melihat apakah itu bekerja lebih baik untuk Anda. Untuk menjalankannya, cukup pilih "Cases → Learn Using Gradient".

Catatan untuk Siswa dari Learning Theory:

Algoritma Gradient Descent yang digunakan di Netica adalah "kemiringan gradien konjugasi/ mentafsirkan (conjugate gradient descent)" untuk memaksimalkan probabilitas data, memberikan network, dengan menyesuaikan entri Tabel CP. Secara umum algoritma ini menyatu lebih cepat daripada EM learning, namun mungkin lebih rentan terhadap local maxima.

Komentar yang diundang

Didalam Norsys, kita menghargai komentar yang ada pada algoritme pembelajaran lanjutan/ advanced learning algorithms kita, dan laporan tentang keberhasilan atau kegagalan tertentu yang Anda dapatkan dari kumpulan data Anda. (email info@norsys.com)

2. Menguji Nets dengan Cases

Tujuan dari tes ini adalah untuk menilai belief network dengan menggunakan seperangkat cases nyata untuk melihat seberapa baik prediksi atau diagnosis kecocokan sesuai dengan cases sebenarnya. Ini bukan untuk decision networks.

Tes ini memungkinkan Anda melihat kelemahan di network Anda. Dengan itu Anda dapat menemukan node-node yang ramalannya berkorelasi lemah dengan kenyataan. Anda mungkin ingin memeriksa ulang tabel probabilitas kondisional mereka, menyediakan data tambahan untuk learning, atau tidak memanfaatkan ramalan mereka sampai kinerjanya membaik.

Ide dasar dari tes ini adalah bahwa kita membagi node net menjadi dua kelas:

- Observed (diamati)
- Unobserved (tidak teramati)

Node yang observed akan "diamati" dalam arti bahwa mereka akan memiliki nilai yang terbaca dari case file. Nilai yang diamati ini kemudian digunakan untuk memprediksi nilai node yang tidak diobservasi/ unobserved nodes dengan menggunakan bayesian belief updating.

Proses ini diulang untuk setiap cases dalam case file. Untuk setiap cases tersebut, kita membandingkan nilai prediksi untuk unobserved nodes dengan yang benar-benar diamati dalam case file. Kita mencatat semua keberhasilan dan kegagalan. Statistik ini dikumpulkan dan dipresentasikan dalam laporan akhir yang dideskripsikan, untuk setiap unobserved nodes, seberapa baik kinerjanya, yaitu seberapa sering prediksi tersebut akurat.

Prosedur untuk melakukan tes adalah sebagai berikut:

1. Pilih node yang Anda tidak berharap network mengetahui nilai selama inferensinya. Ini adalah node yang tidak teramati/ unobserved nodes. Misalnya, jika network untuk diagnosis medis, Anda dapat memilih disease node/ nodes penyakit dan nodes yang mewakili keadaan internal lain yang tidak dapat diamati. Seringkali Anda tertarik pada bagaimana net berperilaku dalam pengaturan prediksi yang realistis, jadi Anda memilih sebagai apa yang tidak teramati yang ada dalam konteks dunia nyata yang tidak dapat diobservasi. Tapi Anda tidak perlu dibatasi oleh aturan ini. Setiap subset dari node bisa berada diantara node yang tidak teramati. Harus selalu ada setidaknya satu node yang tidak teramati, tentu saja, karena jika tidak, tidak ada yang bisa diprediksi dan tesnya akan menjadi tidak berarti.
2. Pilih Network → "Test Using Cases". Anda akan ditanya case files mana yang akan digunakan, dan setelah Anda memilihnya, Netica akan mulai memproses. Jendela Pesan akan sampai ke depan dan menampilkan sebagian kecil dari cases yang diproses sejauh ini. Tahan tombol CTRL + SHIFT + LEFT BUTTON pada saat yang bersamaan jika Anda ingin menghentikan pemrosesan cases dan mencetak hasil yang diperoleh sejauh ini.

Ketika Netica selesai, ia akan mencetak laporan untuk masing-masing node yang tidak teramati (contoh laporan dan analisisnya muncul di bagian bawah halaman ini). Laporan tersebut mencakup hal-hal berikut:

| | | |
|---|---|--------------------------------------|
| confusion matrix error rate calibration table | logarithmic loss score quadratic (Brier) score spherical payoff score | surprise indexes test sensitivity |
|---|---|--------------------------------------|

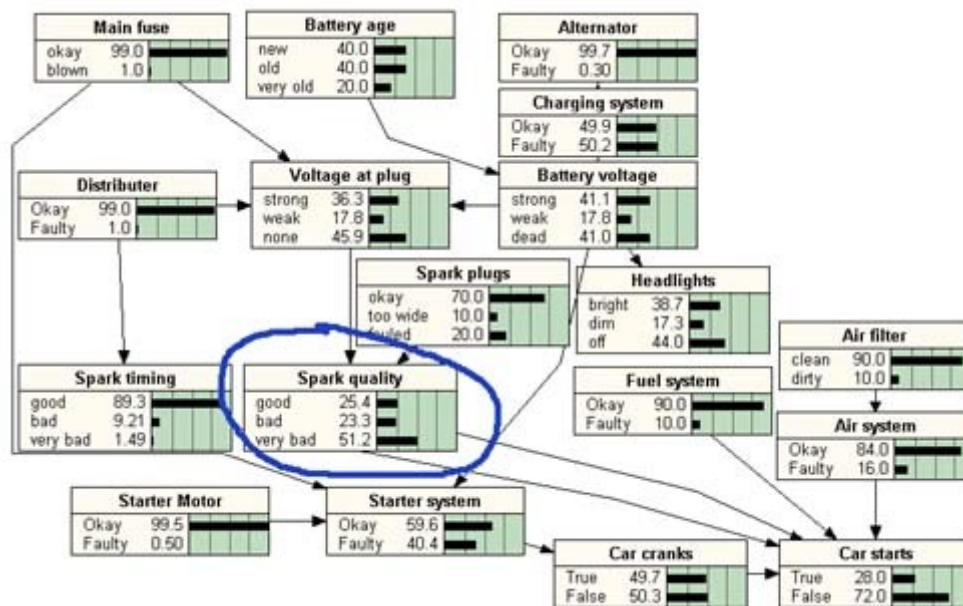
Untuk binary nodes itu juga melaporkan:

| | | |
|------------------|------------------|---------------------------|
| test specificity | predictive value | predictive value negative |
|------------------|------------------|---------------------------|

Hubungi Norsys dengan cara mudah untuk menghasilkan receiver operating characteristic (ROC) plots.

Bagian berikut menjelaskan secara lebih rinci setiap bagian dari laporan. Hal ini dilakukan dalam konteks sample report, untuk kemudahan ilustrasi.

Sample Report



Berikut adalah contoh laporan untuk node yang bernama "SpkQual" (dengan judul node "Spark quality"), dilingkari diatas, di tutorial net CarDiagnosis.dne.

For SpkQual: Spark quality

Confusion:

|Predicted..... | good | bad | very_b | Actual |
|---------------------|------|-----|--------|----------|
| good | 253 | 0 | 0 | good |
| bad | 22 | 176 | 4 | bad |
| very_bad | 13 | 19 | 430 | very_bad |

Error rate = 6.325%

Scoring Rule Results:

Logarithmic loss = 0.2144
Quadratic loss = 0.1099
Spherical payoff = 0.9409

Calibration:

```

good    0-0.5: 0 | 0.5-1: 0 | 1-2: 0 | 2-5: 0 |
        5-80: 49 | 80-95: 87.5 | 95-98: 95.7 |
bad     0-1: 0 | 1-2: 1.52 | 2-5: 2.4 | 5-10: 5.17 |
        10-50: 20 | 50-85: 82.6 | 85-95: 90 | 95-100: 100 |
very_bad 0-0.1: 0 | 0.1-0.5: 0 | 0.5-5: 6.94 | 5-10: 9.33 |
        10-20: 16.2 | 20-95: 83.3 | 95-98: 98.9 | 98-99: 100 |
        99-100: 100 |
Total   0-0.1: 0 | 0.1-0.5: 0 | 0.5-1: 0 | 1-2: 0.431 |
        2-5: 2.5 | 5-10: 6.28 | 10-15: 10.9 | 15-20: 13.3 |
        20-50: 30.1 | 50-80: 81.5 | 80-90: 86 | 90-95: 93.7 |
        95-98: 97.6 | 98-99: 100 | 99-100: 100 |

```

Times Surprised (percentage):

| State | Predicted Probability..... | | | |
|----------|----------------------------|----------------|---------------|-------------|
| | < 1% | < 10% | > 90% | > 99% |
| good | 0.00 (0/312) | 0.00 (0/614) | 6.86 (14/204) | 0.00 (0/0) |
| bad | 0.00 (0/225) | 1.98 (13/657) | 0.00 (0/69) | 0.00 (0/0) |
| very_bad | 0.00 (0/216) | 3.32 (12/361) | 0.25 (1/399) | 0.00 (0/31) |
| Total | 0.00 (0/753) | 1.53 (25/1632) | 2.23 (15/672) | 0.00 (0/31) |

Analisis dari Sample Report

Confusion Matrix/ Matriks Kebingungan. State yang mungkin dari Spark Quality adalah bagus, buruk dan sangat buruk. Untuk setiap cases yang diproses, Netica menghasilkan kepercayaan/ belief untuk masing-masing state ini. State yang paling mungkin (yaitu yang memiliki kepercayaan tertinggi) dipilih sebagai prediksi untuk nilai Spark Quality. Ini kemudian dibandingkan dengan nilai sebenarnya dari Spark Quality untuk cases itu, sehingga case file bisa memasoknya. **Confusion Matrix** menyediakan jumlah total case pada masing-masing dari 9 situasi: (Prediksi = baik, aktual = baik), (Prediksi = buruk, aktual = baik), dll. Jika network berkinerja baik maka entri sepanjang main diagonal akan besar dibandingkan dengan yang tidak.

Error Rate/ Tingkat Kesalahan. Dalam laporan sampel kita, tingkat kesalahan adalah 6,325%. Ini berarti bahwa pada 6,325% cases dimana cases file memberikan/ menyediakan suatu nilai Spark Quality, network memperkirakan nilai yang salah, dimana prediksi tersebut diambil sebagai state dengan keyakinan/ belief tertinggi (sama seperti confusion matrix).

Scoring Rule Results. Skor ini tidak hanya mengambil state yang paling mungkin sebagai prediksi, namun mempertimbangkan tingkat kepercayaan aktual (actual belief levels) dari state dalam menentukan seberapa baik mereka setuju dengan nilai dalam case file. Hasil ini dihitung dengan cara standar untuk aturan penilaian/ scoring rules. Untuk informasi lebih lanjut lihat referensi mengenai aturan penilaian, seperti Morgan & Henrion90 atau Pearl78 .

Logarithmic and Quadratic loss, and Spherical Payoff. Logarithmic loss values dihitung menggunakan log alami, dan berada diantara 0 dan smpa dengan tak terbatas/ infinity inclusive, dengan nol menunjukkan kinerja terbaik. Quadratic loss (juga dikenal sebagai Brier score) adalah antara 0 dan 2, dengan 0 menjadi yang terbaik, dan spherical payoff adalah antara 0 dan 1, dengan 1 adalah yang terbaik.

Persamaan masing-masingnya adalah:

Logarithmic loss = MOAC [- log (Pc)]

Quadratic loss = MOAC [1 - 2 * Pc + sum[j=1 to n] (Pj ^ 2)]

Spherical payoff = MOAC [Pc / sqrt (sum[j=1 to n] (Pj ^ 2))]

Dimana Pc adalah probabilitas yang diprediksi untuk keadaan/ state yang benar, Pj adalah probabilitas yang diprediksi untuk state j, n adalah jumlah state, dan MOAC adalah singkatan dari mean (average) over all cases/ (titik) pertengahan (rata-rata) untuk semua cases (yaitu semua cases dimana case file yang menyediakan nilai untuk node yang dimaksud).

Calibration/ Pencocokan/ Pengujian. Ini menunjukkan apakah confidence/ kepercayaan yang diungkapkan oleh network sesuai (yaitu "dikalibrasi dengan baik"). Misalnya, jika network meramalkan cuaca, Anda mungkin ingin tahu: Dari semua kejadian yang dikatakan 30% kemungkinan hujan, berapa persentase waktu yang terjadi? Jika ada banyak cases, jawabannya harus mendekati 30%. Untuk setiap keadaan node terdapat sejumlah item yang dipisahkan oleh palang vertikal (|). Setiap item terdiri dari rentang persentase probabilitas R, diikuti oleh titik dua (:) dan kemudian single/ satu persentase X. Ini berarti bahwa sepanjang waktu belief/ kepercayaan terhadap keadaan itu berada dalam kisaran R, X persen dari nilai sebenarnya state.

Contohnya
hujan 0-10: 8.5 |

berarti bahwa sepanjang waktu, belief/ kepercayaan akan hujan berkisar antara 0 dan 10%, 8,5% pada saat hujan turun. Alasan mengapa rentang probabilitasnya tidak merata, dan berbeda dari satu state ke state lainnya, dan berjalan untuk berjalan (run to run), adalah bahwa mereka dipilih sehingga persentase X cukup akurat. Ukuran bin harus disesuaikan, atau mungkin tidak ada cukup banyak cases yang jatuh didalam bin itu. Semakin banyak cases yang Anda proses, semakin bagus rentang probabilitasnya. Hasil kalibrasi sering digambarkan sebagai grafik (dikenal sebagai "kurva kalibrasi") dimana kalibrasi ideal adalah garis lurus diagonal. Untuk informasi lebih lanjut, lihat teks yang membahas "kalibrasi" probabilitas misalnya Morgan & Henrion⁹⁰, hal.110, yang disebutkan diatas.

Times Surprised Table. Ini menunjukkan seberapa sering network cukup percaya diri/ confident dalam beliefnya, namun salah. Ada kolom yang menjadi 90% confident dan 99% confident (masing-masing beliefs lebih tinggi dari 90% atau 99%), dan juga 90% dan 99% confident bahwa nilai node tidak akan menjadi keadaan/ state tertentu (yaitu masing masing kurang dari 10% atau 1%). Rasio tersebut menunjukkan berapa kali jumlah kesalahannya salah dari berapa kali prediksi tersebut dibuat dengan tepat, dan persentase juga dicetak. Jika network berkinerja baik, persentase ini akan rendah, namun perlu diingat bahwa sangat masuk akal jika salah dengan prediksi 10% atau 90% prediction 10% dari waktu tertentu, dan menjadi salah dengan angka 1% atau 99% prediksi 1% dari waktu. Jika network jarang membuat prediksi kuat (yaitu kepercayaan/ beliefs jarang mendekati 0 atau 1), maka sebagian besar rasio ini akan menjadi 0/0.

Quality of Test(binary nodes) or Test Sensitivity (nState>2). Laporan ini berguna bila output dari network akan digunakan untuk menentukan suatu aksi, dengan satu aksi yang sesuai dengan setiap keadaan node. Sebagai contoh medis, nodes itu mungkin "Penyakit-A/ Disease-A" dan memiliki kedua state "Ada" dan "Tidak Ada"/ "Present" and "Absent". Jika setelah memperbarui/ updating untuk sebuah cases, network melaporkan "Present", maka perawatan/ pengobatan tertentu akan dimulai, namun jika dilaporkan "Absent" maka perawatan tidak akan dimulai. Pertanyaannya adalah, pada kemungkinan/ probabilitas apa "Present" harus kita katakan bahwa network melaporkan Present?

Confusion matrix dan *error rate* yang dibahas diatas ditentukan dengan menggunakan keadaan kemungkinan maksimum/ *maximum likelihood state* (yaitu yang memiliki belief tertinggi setelah memperbarui/ updating). Untuk variabel biner, ini berarti memilih state pertama hanya jika belief lebih tinggi dari 50%. Tetapi jika masing-masing state memiliki biaya kesalahan klasifikasi yang berbeda, Anda mungkin tidak ingin probabilitas cutoff menjadi 50%. Dalam contoh medis, mungkin berbahaya untuk tidak mengobati pasien yang menderita penyakit ini, tapi tidak terlalu serius jika diperlakukan dengan tidak perlu. Jadi Anda ingin network melaporkan "Present" jika probabilitas penyakit ini di atas sejumlah kecil, seperti 2%. Ini adalah masalah trading off dari tingkat positif palsu/ *false positives* melawan tingkat negatif palsu/ *false negatives*.

Idealnya Anda hanya akan mengubah network ke *decision network*, dengan menambahkan suatu *decision node* untuk aksi yang akan diambil dan *utility node* untuk biaya kesalahan klasifikasi/ cost of misclassification. Namun, pada saat network dibangun dan dinilai sesuai kegunaannya, utilitas mungkin tidak diketahui.

Bagian "*Quality of Test*" memiliki hasil kinerja untuk serangkaian probabilitas cutoff threshold (yang dijalankan secara vertikal pada kolom pertama). Untuk setiap cases, beliefs yang diberikan oleh network diubah menjadi "prediction". Prediksinya adalah "*first state*" jika belief untuk state pertama lebih tinggi daripada probabilitas cutoff, dan "*second state*" jika lebih rendah. Anda mungkin ingin mengubah urutan state, sehingga state pertama adalah state "*positive*", agar sesuai dengan makna konvensional. Makna dari kolom tersebut adalah:

| | |
|---------------------------|---|
| Sensitivity | Of the cases whose actual value was the first state, the fraction predicted correctly/ Dari cases yang nilai aktualnya adalah keadaan pertama, fraksi tersebut diprediksi dengan benar |
| Specificity | Of the cases whose actual value was the second state, the fraction predicted correctly/ Dari cases yang nilai aktualnya adalah keadaan kedua, fraksi tersebut diprediksi dengan benar |
| Predictive Value | Of the cases the network predicted as first state, the fraction predicted correctly/ Dari cases yang diprediksi network sebagai keadaan pertama, fraksi diprediksi dengan benar |
| Predictive Value Negative | Of the cases the network predicted as second state, the fraction predicted correctly/ Dari cases yang diprediksi network sebagai keadaan kedua, fraksi diprediksi dengan benar |

Seringkali data ini dirangkum dengan grafik yang disebut kurva **ROC (receiver operating characteristic/** karakteristik operasi penerima). Untuk menggunakan Excel (tersedia dari Microsoft) untuk membuat kurva ROC dari data ini, pilih keseluruhan tabel (kecuali judul) dan sambil menahan tombol, ketikkan `tcz`. Kemudian buka file Excel yang disebut "Graph_ROC.xls" (tersedia dari Norsys [file downloads](#) directory), tempelkan/paste ke sel yang ditunjukkan, dan grafik akan digambar.

Jika node memiliki lebih dari 2 state, sebaliknya Anda akan mendapatkan bagian "*Test Sensitivity*". Angka pertama dari setiap "kolom" adalah *cutoff threshold probability*. Angka kedua dari setiap kolom adalah jumlah cases yang nilai sebenarnya/ actual value adalah keadaan yang diberikan disisi kiri baris, dan network mana yang diprediksi dengan benar berada dalam keadaan itu (yaitu belief-nya lebih besar daripada probabilitas cutoff), dibagi dengan jumlah total cases yang nilainya sebenarnya adalah keadaan itu.

Kelihatannya aneh bahwa probabilitas cutoff berubah dalam lompatan ukuran aneh. Alasannya adalah bahwa Netica hanya melaporkan nilai yang bisa mengumpulkan cukup data. Jadi menjalankan tes menggunakan sejumlah besar cases umumnya menghasilkan pembagian yang lebih baik dari kolom cutoff.

CATATAN pada Pengujian Net

- Jika Anda memiliki temuan/ findings yang dimasukkan sebelum memilih Network → Test With Cases, mereka akan diperhitungkan selama pembaharuan semua belief (kecuali jika case file memiliki kolom untuk node tersebut). Netica akan memperingatkan Anda dalam event ini, sehingga Anda tidak mendapatkan hasil yang salah dengan secara tidak sengaja meninggalkan beberapa temuan/ finding didalam network. Situasi dimana Anda ingin meninggalkan temuan/ finding didalam network adalah jika network dirancang untuk kelas dari cases yang lebih luas daripada case file. Misalnya, jika Anda memiliki network yang dirancang untuk menangani orang dari kedua tipe kelamin (dan memiliki nodes 'gender'), namun case file hanya berisi perempuan, Anda harus memasukkan temuan/ finding 'perempuan' untuk nodes 'gender' sebelum menilai/ grading network
- Jika temuan/ findings untuk non-unobserved nodes dalam case file tidak mungkin dilakukan menurut network, maka pesan kesalahan temuan yang tidak konsisten akan ditampilkan, cases itu akan diabaikan, dan pemrosesan akan berlanjut. Jika network membuat prediksi untuk unobserved nodes yang tidak sesuai dengan case files, maka tentu saja tidak ada pesan kesalahan yang akan dihasilkan, network hanya akan dinilai lebih buruk (dan memiliki suatu logarithmic loss/ kerugian logaritma dari INFINITY).
- Bergantung pada aplikasi Anda, semua tindakan/ measures yang dihitung bisa menjadi hal yang paling berharga bagi Anda. Namun, jika Anda menginginkan satu nomor untuk menilai network, dan tidak yakin yang mana yang akan dipilih, kita menyarankan logarithmic loss.
- Fungsi ini akan dengan benar mendukung kolom 'NumCases' pada case files, jika ada.
- Serta menilai suatu network, fitur ini juga bisa digunakan untuk mengetahui nilai dari kombinasi findings nodes dilingkungan dunia nyata. Dengan memilih node tambahan/ extra nodes pada langkah pertama, Anda dapat membuat beberapa kemungkinan temuan dari case files yang tidak tersedia untuk network. Maka Anda bisa melihat seberapa besar hasil dari network yang terdegradasi dengan tidak memiliki akses terhadap temuan tersebut. Dalam contoh medis yang disebutkan sebelumnya, Anda mungkin juga memilih 'Blood Test' dan 'Smear Test', dan kemudian membandingkan confusion matrix baru yang dihasilkan dengan yang lama, untuk mengetahui apakah jumlah false negatives dan false positives dari penyakit serius berubah secara signifikan.
- Fitur ini juga tersedia untuk programmer yang menggunakan Netica API; hubungi Norsys untuk informasi lebih lanjut.

Dokumentasi yang lebih lengkap akan tersedia didalam Net Testing dikemudian hari. Jika ada pertanyaan, saran, atau permintaan, atau jika Anda menemukan masalah dengan perangkat lunak yang ada, tolong email Norsys.

3. Mengelola Cases Files

Cases adalah atom dari pengalaman, dasar dari learning, dan kadang berguna untuk mengolah atom-atom ini dengan cara-cara khusus. Selama bertahun-tahun, Netica menjadi semakin kuat, dan fleksibel dalam menangani cases. Kita berharap evolusi ini berlanjut. Jika Anda memikirkan cara baru untuk mengelola cases yang akan mempermudah pekerjaan Anda, beri tahu kita.

Membaca cases satu per satu. Kita telah melihat bagaimana memilih Relation → "Incorp Case File" akan membaca semua cases dalam file dalam one pass. Kadang-kadang sangat mudah untuk membaca cases secara tunggal, untuk melihat bagaimana probabilitas network berkembang saat learning berlangsung, atau bahkan hanya sebagai cara melihat setiap cases dalam instantiasinya. Untuk mulai membaca cases, pilih : File → "Get Case" atau tekan tombol F8.

Kunci F8 dan File → "Get Case", bagaimanapun, memiliki perilaku yang sedikit berbeda:

1. File → "Get Case" selalu memunculkan dialog yang menanyakan case file mana yang akan Anda baca, dan juga cases mana yang harus dibaca. Jika case file menyimpan IDnums (lihat [Case File Format](#)), dialog akan menanyakannya, jika tidak, akan menanyakan posisi cases ini.
2. Kunci F8 melakukan hal yang sama seperti diatas, tapi hanya jika pembacaan single cases belum dimulai. Setelah itu, secara otomatis akan membaca cases berikutnya didalam file (yang langsung mengikuti yang terakhir dibaca), tanpa memunculkan kotak dialog apapun. Ini membuatnya ideal untuk "melangkah melalui" cases, dan biasanya bagaimana Anda akan menggunakan kunci ini. Perhatikan bahwa Netica mencetak nomor cases di jendela Messages, sehingga Anda selalu dapat melihat ke sana untuk melihat cases mana yang terakhir dibaca.

Menerapkan cases dari satu network ke network lainnya. Sangat dapat diterima untuk membaca sebuah cases ke dalam network A, ketika cases itu dibuat dan disimpan dari network lain, B. Tidak perlu A dan B berbagi node yang sama. Namun, hanya node di B yang memiliki nama identik, nomor state identik, dan nama state yang identik (jika nama state ada) dengan yang ada di A, akan terpengaruh. Semua yang lainnya diabaikan. Perhatikan bahwa semua perbandingan nama bersifat case-sensitive.

Nilai numerik lebih diutamakan daripada nilai-nilai state. Nilai bilangan asli yang Anda masukkan sebagai temuan/ findings untuk discretized continuous nodes disimpan dalam case files, bukan hanya state yang sesuai dengan mereka. Hal ini memungkinkan pembacaan cases ke network lain yang node untuk variabel tersebut telah diskretisasi dengan cara yang berbeda, katakanlah dengan lebih atau sedikit state.

Nilai state yang hilang dan state "Lainnya". Jika Anda membaca dalam sebuah cases, dan case files memiliki nilai yang bukan merupakan state dari node yang sesuai didalam network, maka pesan kesalahan akan ditampilkan. Misalnya, jika node 'warna' didefinisikan hanya dengan dua state, 'merah' dan 'hijau', dan dalam cases file nilai 'biru' ditemukan, maka pesan kesalahan akan ditampilkan. Namun, ada pengecualian terhadap aturan ini. Jika node (dalam network, bukan case files) memiliki state bernama 'other' atau 'Other'. Maka case akan terbaca tanpa kesalahan, dan menemukan node-nya akan menjadi 'other'.

NETICA CASE FILE FORMAT

2002-11-02

Copyright 2002-2004 by Norsys Software Corp.

Structure

Case files (single-case atau multi-case) adalah file teks ASCII murni. Mungkin berisi "// ~->[CASE-1]->~" disuatu tempat di 3 baris pertama. Kemudian muncul garis yang terdiri dari judul/ heading untuk kolom. Setiap heading sesuai dengan satu variabel dari cases ini, dan merupakan nama dari node yang digunakan untuk mewakili variabel (kadang-kadang variabel disebut atribut dan entri dalam nilai kolom, yaitu nilai atribut/ attribute-value). Headings dipisahkan oleh spasi dan/ atau tab (tidak masalah berapa banyak).

Data case berikutnya, dengan satu case per baris (single-case files saja satu garis seperti itu). Nilai variabel dalam urutan yang sama dengan judul baris, dan dipisahkan oleh spasi atau tab (kolom tidak memiliki "line-up" seperti yang mereka lakukan pada contoh dibawah ini).

Discrete Variables

Nilai dari suatu variabel diskrit diberikan oleh nama state, atau oleh nomor statenya didahului oleh karakter '#' (state pertama adalah # 0). Menggunakan nama state lebih diutamakan, karena tatanan/ urutan state dapat diubah kapan, dan itu akan membuat sebuah file dengan nomor state tidak valid. Simbol '#' adalah disarankan, tapi mungkin dihilangkan jika node tidak memiliki rentang atau nilai yang ditentukan.

Continuous Variables

Nilai dari suatu variabel kontinyu diberikan oleh bilangan dalam bilangan bulat, desimal, atau notasi ilmiah (misalnya -3.21e-7). Jika sudah discretized, kalau begitu nilainya mungkin diberikan oleh nama state atau nomor state sebagai gantinya, tapi nomor kontinyu lebih disukai jika tersedia. Begitulah case filesnya dapat digunakan untuk discretizations variabel yang berbeda di masa depan. Sebaiknya jika nilainya memiliki jumlah angka signifikan yang benar, karena Netica versi masa depan mungkin menggunakan informasi ini.

Missing Values

Jika nilai beberapa variabel tidak diketahui untuk beberapa cases, maka tanda bintang * dimasukkan ke dalam file dan bukan nilainya. Ini adalah dikenal sebagai "missing data".

Comments

Mungkin ada banyak ruang atau tab pada akhir baris seperti yang diinginkan, dan mungkin juga ada C / C ++ / Java style comments (misalnya garis miring ganda "/*", diikuti teks).

IDnum

Ada dua kolom khusus yang mungkin dimiliki file yang tidak sesuai untuk node. Seseorang memberikan nomor identifikasi untuk setiap cases, yang harus menjadi bilangan bulat antara 0 dan 2 miliar. Judul untuk kolom ini adalah "IDnum". Nomor identifikasi tidak harus dilakukan melalui file. Simbol data yang hilang, *, tidak boleh muncul di kolom ini.

NumCases

Kolom khusus lainnya memiliki judul "NumCases", dan menunjukkan frekuensi atau multiplisitas dari case ini. Banyaknya/ multiplicity dari M menunjukkan M cases dengan nilai variabel yang sama. Hal ini tidak diperlukan untuk menjadi bilangan bulat, sehingga bisa digunakan untuk merepresentasikan frekuensi kejadian jika diinginkan. Simbol data yang hilang, *, tidak boleh muncul di kolom ini.

Examples

1. Berikut adalah daftar "asia.cases". Ini hanya melibatkan discrete nodes dengan nama state, dan memiliki kolom IDnum, namun tidak ada kolom frekuensi.

// -->[CASE-1]->~

| | | | | | | | | |
|-------|-----------|--------------|-----------|---------|--------|----------|------------|---------|
| IDnum | VisitAsia | Tuberculosis | Smoking | Cancer | TbOrCa | XRay | Bronchitis | Dyspnea |
| 1 | No_Visit | Present | Smoker | Absent | True | Abnormal | Absent | Present |
| 2 | No_Visit | Absent | Smoker | Absent | False | Normal | Present | Present |
| 3 | No_Visit | Absent | Smoker | Present | True | Abnormal | Present | Present |
| 4 | No_Visit | Absent | NonSmoker | Absent | False | Normal | Absent | Absent |
| 5 | No_Visit | Absent | Smoker | Present | True | Abnormal | Present | Present |
| 6 | No_Visit | Absent | Smoker | Absent | False | Abnormal | Present | Present |
| ... | | | | | | | | |
| 119 | No_Visit | Absent | Smoker | Absent | False | Normal | Present | Present |
| 120 | No_Visit | Absent | Smoker | Present | True | Abnormal | Present | Present |

2. Berikut adalah contoh lain dari sebuah case files, kali ini untuk mobil yang dibawa ke garasi. Ini memiliki discrete dan continuous variables, nomor state dan nama state, dan tanda bintang untuk entri yang hilang.

// -->[CASE-1]->~

| | | | | | | | | | | | |
|--------|--------|--------|--------|---------|--------|-------|-------|---------|------|----------|--------|
| Starts | BatAge | Cranks | Lights | StMotor | SpPlug | MFuse | Alter | BatVolt | Dist | PlugVolt | Timing |
| False | 5.9 | False | #0 | * | fouled | okay | * | dead | * | * | good |
| False | 1.3 | False | #0 | * | okay | okay | * | dead | * | none | bad |
| False | 5.2 | False | #0 | Okay | okay | okay | Okay | dead | Okay | none | good |
| True | 4.1 | True | #2 | * | okay | okay | * | strong | Okay | strong | * |
| True | 2.7 | * | #2 | * | wide | okay | * | strong | Okay | * | * |
| * | * | True | #2 | * | fouled | okay | * | * | Okay | strong | good |
| False | 1.7 | True | #0 | Okay | okay | okay | Okay | dead | * | none | good |
| True | 2.9 | True | #2 | * | * | * | * | strong | Okay | strong | * |

Future

Versi masa depan Netica akan mendukung operasi yang lebih maju dengan cases, termasuk representasi file yang lebih efisien, dan cara menggunakan belief network sebagai "fungsi pengindeksan" untuk melakukan tipe pencarian umum didalam case-based reasoning. Namun, tipe format file yang dijelaskan diatas akan selalu didukung juga.

Dynamic Bayes Nets

Dynamic Bayes nets juga dikenal sebagai DBN atau temporal Bayes nets. Mereka memungkinkan Anda menentukan Bayes net model yang memiliki "waktu tunda/ *time delay*" links, yang menunjukkan bahwa nilai node anak bergantung pada nilai induk pada waktu sebelumnya. Kemudian Anda memperluas DBN sehingga beberapa node didalam net asli menjadi beberapa node didalam net yang diperluas, yang menunjukkan nilai dari variabel tersebut pada berbagai titik/ poin waktu.

DBN dapat memiliki siklus terarah/ *directed cycles*, selama ada link tunda/ *delay link* disuatu tempat disepanjang setiap siklus. *Delay link* dapat digunakan untuk memodelkan umpan balik/ *feedback*. Setelah diperluas, ia tidak lagi memiliki siklus atau link penundaan/ *cycles* atau *delay links*.

Kami meningkatkan kemampuan DBN Netica pada setiap rilis, jadi jika Anda bekerja dengan DBN, Anda harus menggunakan setidaknya versi 5.02. Download versi terbaru dari situs ftp kami.

Biasanya paling mudah bekerja melalui contoh DBN dulu. Bayes nets disebut "Bouncing", di folder "Examples" Netica, cocok untuk tujuan itu.

Langkah-langkah untuk bekerja dengan Dynamic Bayes Net:

1. Buat DBN/ Create DBN
2. Membangkitkan ekspansi waktu/ Generate time expansion
3. Kompilasi dan gunakan/ Compile and use

Jika Anda memiliki permintaan atau saran untuk fitur DBN Netica, pastikan untuk menghubungi tim dukungan kami, karena kami secara aktif memperbaiki fitur ini.

Lihat juga: Tutorial tentang Contoh Bouncing DBN

1. Buat DBN/ Create DBN

Netica memiliki kemampuan DBN yang unik yang tidak tersedia di software lainnya. Link yang berbeda mungkin memiliki *time delays* yang berbeda, sehingga bila jaringan diperluas, *time slices*/ irisan waktu mungkin memiliki struktur yang berbeda satu sama lain. Dalam net yang diperluas, Netica akan mereplikasi node pada frekuensi yang sesuai untuk memodelkan situasi dinamis yang dimodelkan oleh DBN. Beberapa node hanya akan muncul sekali didalam net yang diperluas (sesuai dengan variabel yang nilainya tidak berubah dari waktu ke waktu), beberapa node akan muncul beberapa kali/ *few times* (untuk variabel yang berubah secara perlahan), dan yang lainnya akan muncul berkali-kali/ *many times* (untuk perubahan variabel yang cepat).

Saat memasukkan jumlah penundaan/ *delay* untuk sebuah link, Anda dapat memasukkan angka (yang paling umum adalah hanya 1), atau nilai dari constant node, atau persamaan berdasarkan nilai satu atau lebih constant nodes. Link dengan *time delays* ditampilkan dalam warna coklat kemerahan.

Untuk mengatur *time delay* beberapa link ke nilai yang sama, pilih linknya, dan pilih Modify → Delay Links, atau klik kanan selection/ pilihan dan pilih Delay. Anda kemudian akan memiliki kesempatan untuk memasukkan *delay* yang Anda inginkan untuk dimiliki semua. Jika Anda membuat *delay* 0, mereka akan menjadi reguler links.

Cara lain untuk mengatur link delays adalah dari node properties dialog. Pilih Delay dari pemilih multiguna/ multipurpose selector dibagian bawah, dan Anda bisa memasukkan jumlah *delay* untuk setiap link yang dimasukkan ke node.

>> Langkah selanjutnya

2. Membangkitkan Ekspansi Waktu/ Generate Time Expansion

Buatlah time expanded version/ versi waktu yang diperluas dari net dengan memilih Network → Expand Time.

Ini akan membuat jendela baru dengan net baru didalamnya. Anda mungkin ingin mengubah ukuran jendela ini agar lebih lebar.

Net baru adalah Bayes net biasa dengan masing-masing Posisi/ Position dan Kecepatan/ Velocity node yang mewakili posisi dan kecepatan pada titik/ poin waktu yang baru, dengan node ke kanan sesuai dengan waktu selanjutnya.

Langkah sebelumnya << >> Langkah selanjutnya

3. Kompilasi dan Gunakan/ Compile and Use

Kompilasi Bayes net untuk kesimpulan/ inferensi probabilitas dengan Network → Compile. Aktifkan automatic updating, jika belum, dengan Toggling Network → Automatic Updating, jadi item menu ditandai dengan tanda centang. Percobaan dengan menetapkan position or velocity/ posisi atau kecepatan (dengan mengklik interval yang diinginkan) untuk menunjukkan pengamatan pada waktu-waktu tertentu, dan melihat bagaimana beliefs terhadap posisi dan kecepatan pada saat lain berubah.

Anda juga dapat mencoba memasukkan beberapa *negative findings* dengan menahan tombol shift saat Anda mengklik interval.

Ingat bahwa hasil numerik tidak akan tepat, karena diskretisasi dan kesalahan sampling/ sampling error dalam mengubah/ mengkonversi persamaan ke tabel probabilitas. Anda mungkin juga ingin mencoba diskritisasi yang lebih baik dengan mengubah Discretization dari net yang tidak dieksploitasi sebelumnya.

4.3 Alat Pemodel DBN

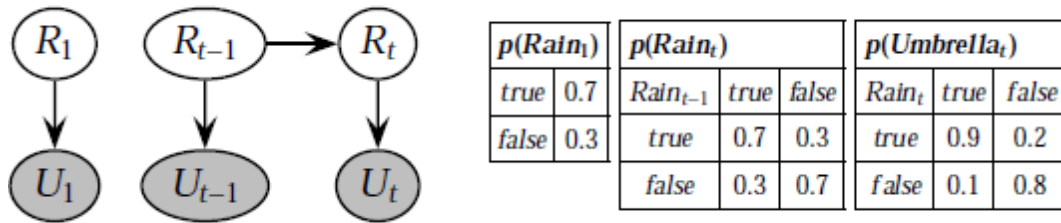
Saat ini, hanya ada dua alat pemodelan yang mendukung penalaran temporal dan memiliki GUI: BayesiaLAB dan Netica. Untuk menggambarkan penggunaan DBN dalam program ini, Umbrella DBN yang juga berfungsi sebagai contoh ilustratif di [RN03] diimplementasikan:

Contoh (Umbrella). Bayangkan seorang penjaga keamanan di beberapa instalasi bawah tanah rahasia. Penjaga itu ingin tahu apakah hari ini hujan, tapi satu-satunya akses ke dunia luar terjadi setiap pagi saat dia melihat direktur masuk, dengan atau tanpa, sebuah payung.

Untuk setiap hari t , himpunan evidence/ bukti berisi satu variabel $Umbrella_t$ (mengamati payung) dan himpunan variabel yang tidak teramati/ unobservable mengandung $Rain_t$ (apakah sedang hujan).

Jika hari ini hujan tergantung pada apakah hujan turun beberapa hari sebelumnya. Model ini dianggap sebagai *first-order Markov* dan *stationary*, jadi $Rain_t$ hanya bergantung pada $Rain_{t-1}$.

Jaringan awal yang dihasilkan, 2TBN dan CPT ditunjukkan pada gambar 4.1.



Gambar 4.1 The umbrella DBN and its CPTs.

4.3.2 Netica

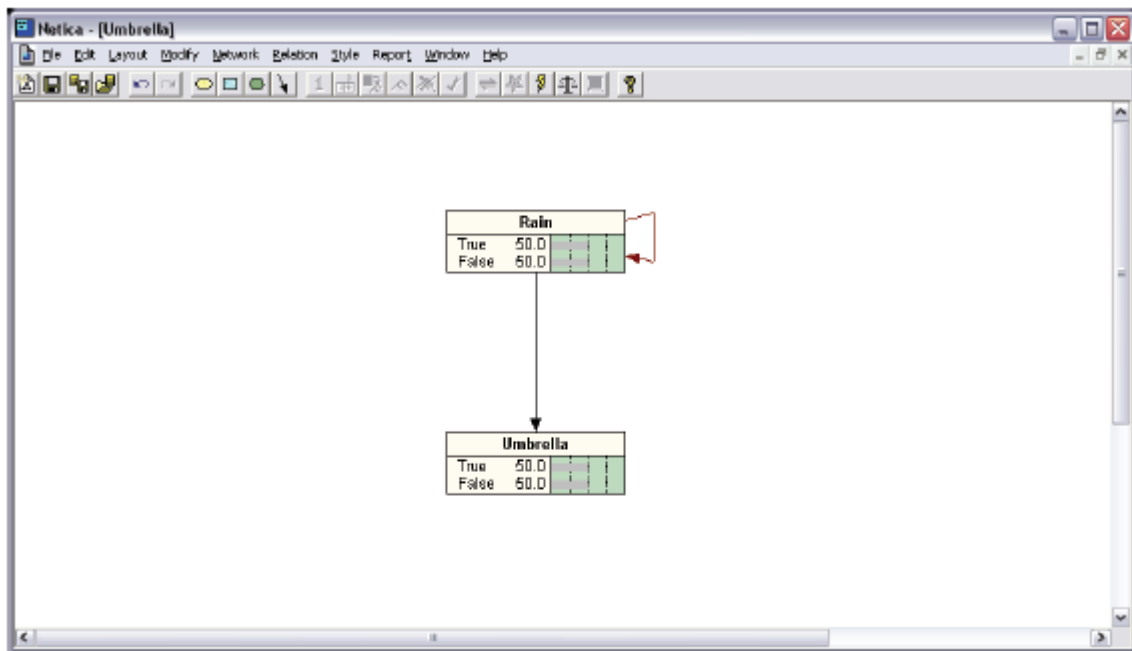
Netica adalah produk dari Norsys (<http://www.norsys.com>) [Nor97], yang terletak di Kanada. Didalam Netica, pengguna merancang (D)BN dan kemudian mengkompilasinya, setelah itu inferensi yang mungkin dilakukan. Kompilasi (D)BN pada dasarnya berarti dikonversi menjadi *junction tree representation*.

Saat merancang DBN, busur temporal/ *temporal arcs* dapat ditambahkan diantara nodes (gambar 4.3). *Temporal arcs* memiliki warna merah. Perbedaannya dengan BayesiaLAB adalah bahwa kita secara eksplisit mendefinisikan hubungan temporal/ temporal relationship (disebut *time-delay* in Netica) dari dua node disalah satu node, bukan pendekatan implisit dari BayesiaLAB.

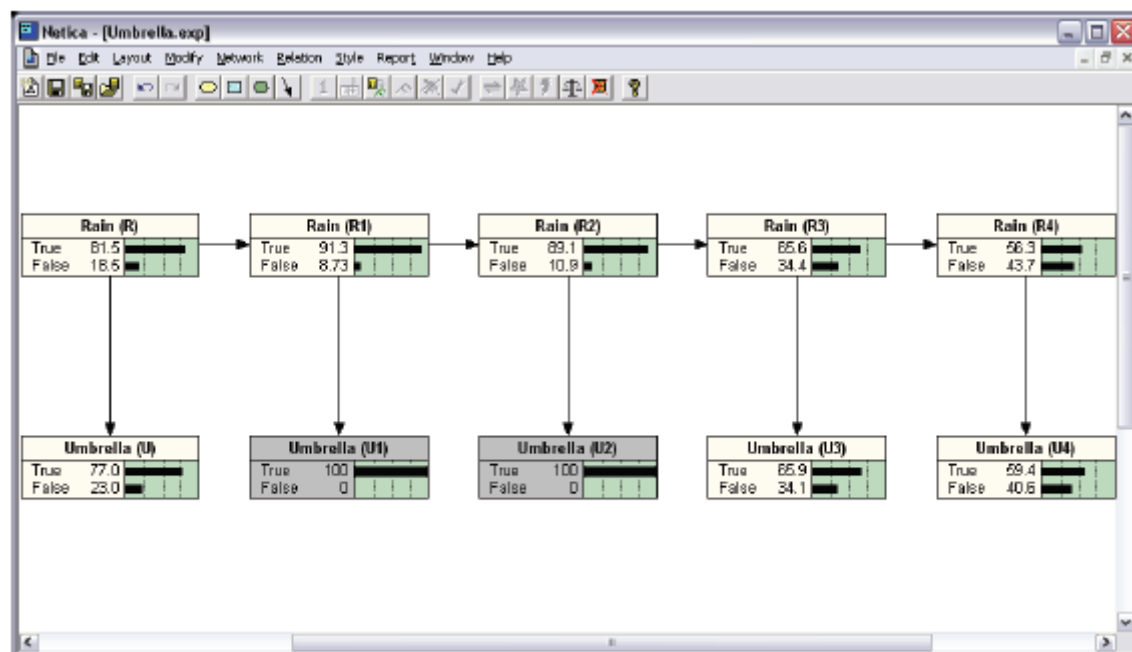
Dengan cara ini, *k-order Markov processes* dapat dimodelkan. Ketika DBN memiliki node dengan hubungan temporal dengan dirinya sendiri (yang biasanya terjadi), definisi DBN berisi loop. Tentu saja, ini hanya berlaku untuk definisi DBN; versi unrolled tidak boleh berisi loop.

Setelah menentukan DBN, dapat dibuka/ unrolled untuk irisan $t/ t \text{ slices}$ dan dikompilasi. BN yang dihasilkan dibuka dijendela baru (gambar 4.3). Inferensi dapat dilakukan pada BN yang dibuka dan dikompilasi ini. Semua metode inferensi yang dibahas di 3.2.2³ didukung. Kita bisa memasukkan bukti/ evidence dengan tangan atau dengan mengimpor suatu data file.

³*filtering, prediction, online/offline smoothing and Viterbi*



(a)



(b)

Gambar 4.3: GUI Netica: (a) *design mode* dimana pengguna merancang BN. (b) *unrolled and compiled (static) BN*/ BN yang dibuka dan dikompilasi (statis) dimana inferensi yang mungkin dilakukan.

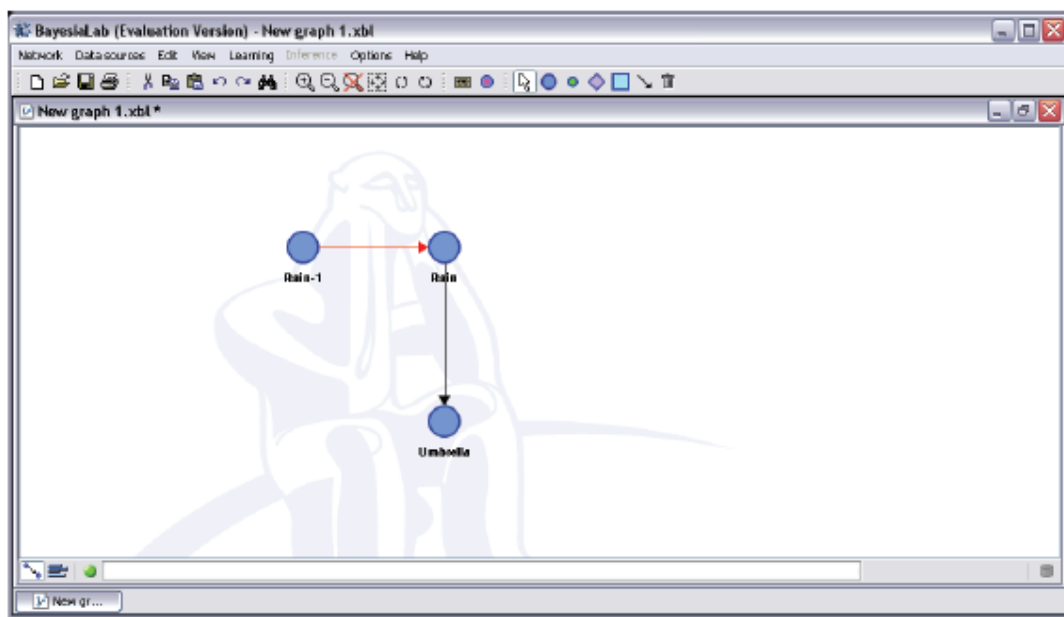
4.3.1 BayesiaLab

BayesiaLab adalah perangkat lunak BN yang dikembangkan oleh perusahaan Perancis Bayesia (<http://www.bayesia.com>) [Bay04]. BayesiaLAB memiliki dua mode:

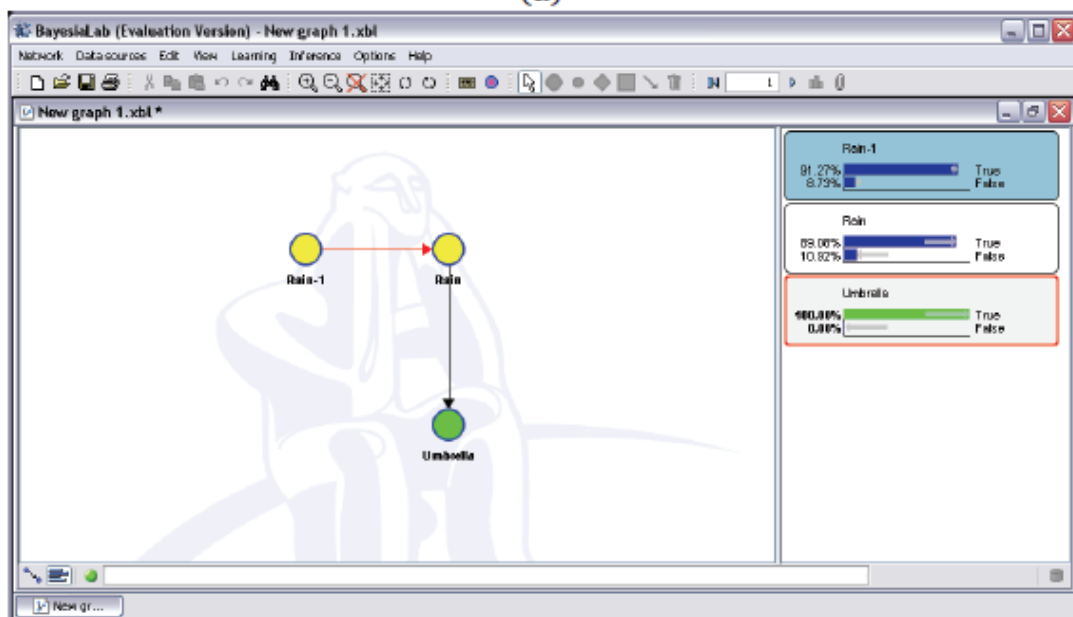
- *modeling mode* untuk merancang BN (gambar 4.2a) dan
- *validation mode* (gambar 4.2b) untuk inferensi/ kesimpulan.

Dalam *modeling mode*, adalah mungkin untuk menambahkan busur temporal/ temporal arcs ke BN untuk menunjukkan bahwa node induk dari dua nodes berada di potongan waktu/ time slice sebelumnya. Hanya *first-order Markov models* yang bisa dirancang seperti ini. Temporal arcs memiliki warna yang berbeda (merah). Pengguna perlu menentukan keadaan awal dan probabilitas temporal/ *initial state* dan *temporal probabilities* dari node. Setelah spesifikasi, *validation mode* dapat dipilih untuk mengikuti perubahan sistem dari waktu ke waktu dengan menjelajah diantara langkah waktu (hanya forward steps/ langkah maju yang didukung) atau dengan menetapkan jumlah langkah dan kemudian merencanakan grafik dari variabel (gambar 4.2c). DBN tidak membuka gulungan secara grafis, hanya nilai yang berubah. Hal ini dimungkinkan untuk membuat bukti/ evidence dengan tangan atau dengan mengimpor data file. Dari metode inferensi yang dibahas di 3.2.2², hanya penyaringan dan prediksi (filtering and prediction) yang mungkin dilakukan.

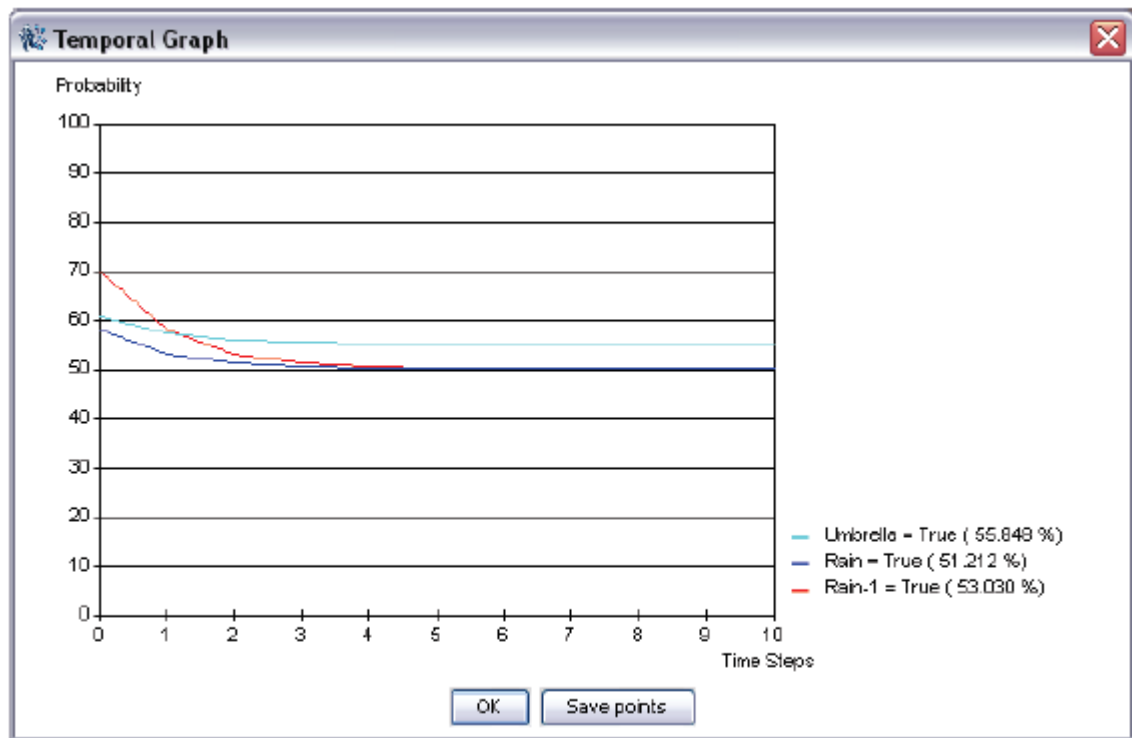
²filtering, prediction, online/offline smoothing and Viterbi



(a)



(b)



(c)

Gambar 4.2: GUI BayesiaLAB: (a) *design mode* dimana pengguna merancang (D)BN. (b) *validation mode* dimana dimungkinkan untuk melakukan inferensi pada (D)BN. (c) *time-graph* dari variabel. Karena tidak ada bukti/ evidence yang ditetapkan, sistem menuju ekuilibrium/ keseimbangan/ kesetimbangan.

4.4 Ringkasan

Setelah melihat dan bermain dengan perangkat lunak yang ada, kita hanya dapat menyimpulkan bahwa tidak ada solusi pemodelan yang memuaskan dengan DBNs. Skrip yang tidak praktis untuk menerapkan DBN, definisi DBN konservatif, kurangnya fungsi impor/ ekspor untuk model BN, dan implementasi yang lambat adalah fitur yang berlaku untuk satu atau lebih paket yang diulas. Paket dengan fungsi paling banyak adalah BNT, tapi sayangnya sangat lambat, tidak memiliki GUI dan memiliki definisi DBMS yang sangat konservatif. Namun, kedua alat pemodelan yang memiliki GUI juga tidak memuaskan, karena mereka juga kekurangan banyak fungsi dan sangat lambat, membuatnya hanya berguna untuk masalah mainan/ toy problems.

Hal yang menyenangkan tentang SMILE dan GeNIe adalah menggabungkan dua hal terbaik: SMILE API menyediakan peneliti dengan semua fleksibilitas dan fungsionalitas yang dia inginkan dan GeNIe GUI membuat fungsionalitas inti relatif mudah diakses oleh publik. Namun, SMILE dan GeNIe perlu diperluas dengan penalaran temporal sebelum dapat digunakan untuk pemodelan dengan DBNs.

Memperluas SMILE dan GeNIe dengan fungsionalitas penalaran temporal tidak sepele. Karena tidak ada pendekatan standar untuk mewakili DBN, ekstensi terdiri dari dua bagian: merancang pendekatan pemodelan, dan menerapkannya dalam perangkat lunak. Pendekatan pemodelan yang dirancang akan dibahas pada bab 7. Perancangan dan implementasi penalaran temporal di SMILE dan GeNIe akan dibahas pada 8.