

RANGKUMAN PRAKTIS

Intelijensi Buatan

Topics

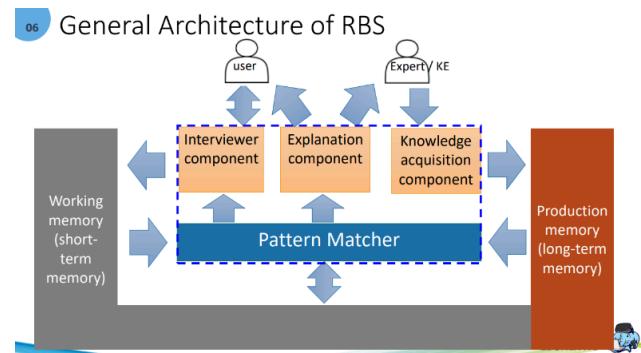
1. Rule Based System
2. Intro to Learning and Data Science
3. EDA
4. Supervised Learning: KNN & Naive Bayes
5. Unsupervised Learning
6. Reinforcement Learning
7. Probabilistic Reasoning System

Bab 4: Rule-Based System

What and Why

Basically KBS with rule sebagai representation knowledge-nya

Simple and most widespread solution



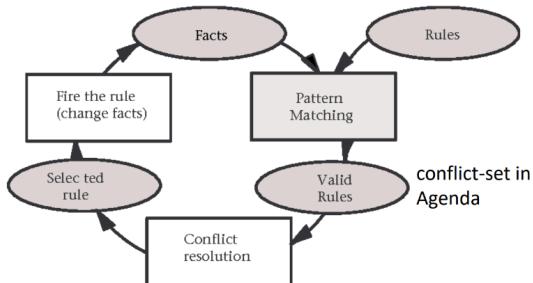
Pattern Matching: Example

IF: is-a (x, horse),
is-parent-of(x, y),
is-fast(y)
THEN: x is valuable

Facts		
Comet	is-a	horse
Prancer	is-a	horse
Comet	is-parent-of	Dasher
Comet	is-parent-of	Prancer
Prancer	is	fast
Dasher	is-parent-of	Thunder
Thunder	is	fast
Thunder	is-a	horse
Dasher	is-a	horse

Forward Chaining

Forward Chaining: Recognize-Act Cycle



Conflict resolution strategy

Global control

1. Selection by order (rule order vs fact recency)
2. Refractoriness (once only)
3. Specificity (paling spesifik, fit ke struktur)

Local control

1. Selection by priority
2. Selection by meta rules

Backwards Chaining

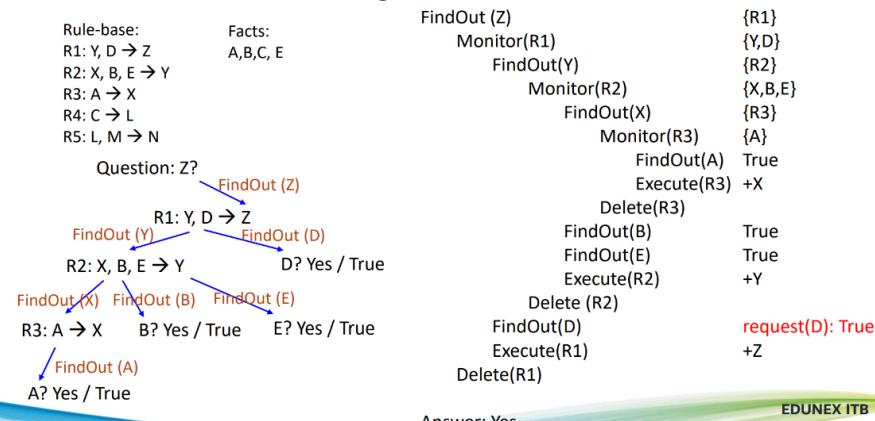
Goal-driven

Jadi pakai goal, nah goal-nya dicari, bisa dari rule apa aja yang ngehasilin goal itu

Untuk semua fakta yang dibutuhin di goal, bisa cari lagi dia dihasilkan oleh rule apa, kalau dia termasuk dalam fakta, true, kalau dia tidak ada di rule atau fakta, false

08

Backward Chaining Process



FindOut itu buat cek dia ada di fakta (pakai **Request**) atau di rule mana, **Monitor** itu buat cek si rule itu apa aja parameternya.

Execute itu buat eksekusi rule yang terpenuhi semua faktanya, buat ngehasilin hasilnya Delete kalau udah di-execute atau ga bisa di-execute.

RBS Features

- Modularity
 - Antara rule itu independen
- Incrementality
 - Bisa ditambah-tambahin rule baru
- Modifiability
 - Rule lama bisa berubah, karena ga ada yang dependen ke dia
- Transparan untuk support system\

Bab 5: Learning

Why learning?

- Jika belum tau environment nya
- Buat konstruksi sistem
- Modifikasi agent biar + performance

Basically learning itu untuk adaptasi agar lebih efisien

- Komponen kinerja apa yang mau dipelajari
- Ada prior knowledge tidak
- Feedback apa yang tersedia untuk learning
- Representasi apa yang digunakan untuk komponen

Tipe Learning

1. Unsupervised (no feedback)
2. Supervised (dikasih contoh i/o pairs)
3. Reinforcement (dikasih reward/punishment)

Supervised

Idenya menggunakan inductive learning, dikasih set of I/O pairs, terus bikin hampiran fungsi target, sebuah hipotesis.

Bisa aja salah karena dia induksi. Tetapi hasil induksi ini yang bakal jadi hipotesis yang digunakan untuk prediksi.

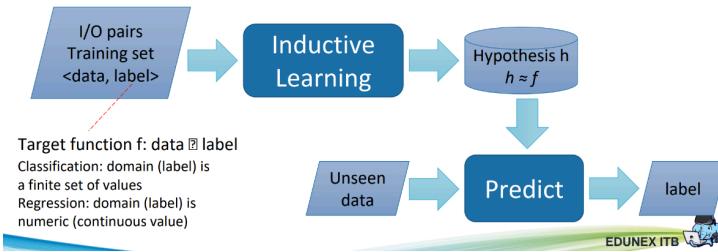
I/O pairs itu <input data, output label>.

Klasifikasi domain labelnya finite set of value, kalau regresi itu domain labelnya numerik (nilai kontinu)

03

Supervised Learning

Learning a (possibly incorrect) general function from specific input-output pairs is called inductive learning



Dataset configuration

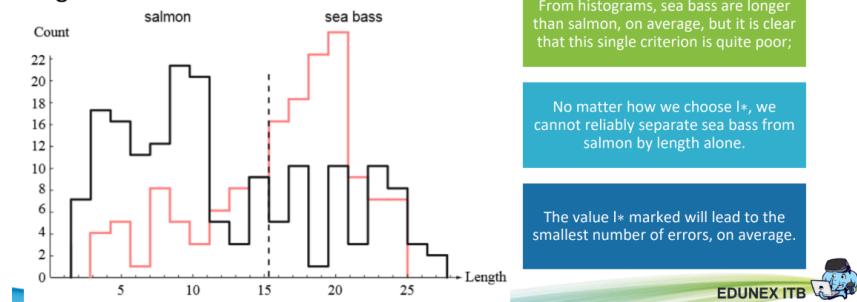
- Enhancement
- Segmentation
- Resize

Contoh model

1. Menggunakan length

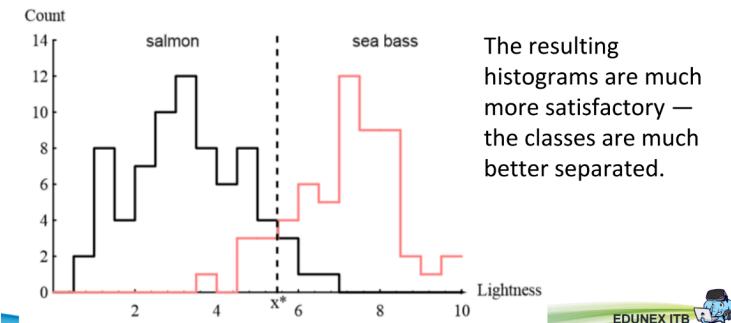
Tentative Model: Length Feature

Suppose somebody at the fish plant tells us that a sea bass is generally longer than a salmon.



2. Menggunakan lightness

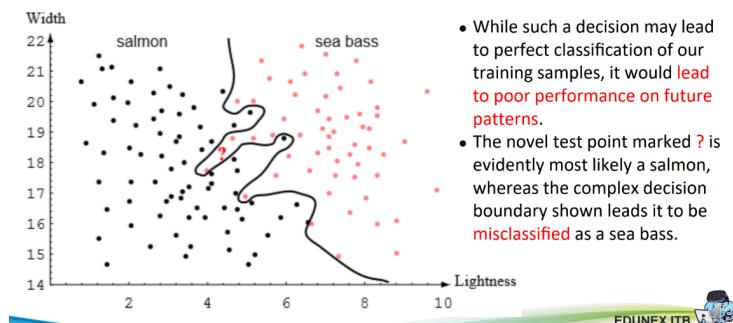
Tentative Model: Lightness Feature



Tetapi sangat mungkin untuk terjadi misklasifikasi

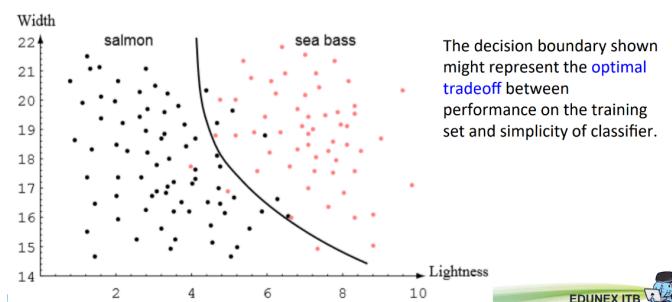
Kalau misal dikasih trainingnya juga akurasinya 100%, nanti malah jelek buat prediksinya, jadi lebih banyak misklasifikasi nantinya (overfit)

Tentative Complex Model: Training Accuracy 100%



Jadinya kita kasih tradeoff performance buat training set biar lebih simpel classifier nya, yang bakal bikin jadi lebih bagus buat prediksi

Optimal Model: Better Generalization



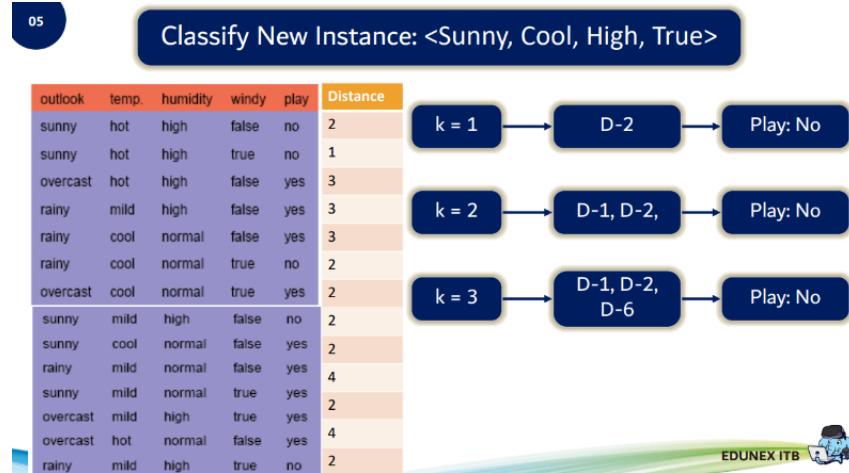
K-Nearest Neighbor

Supervised learning, instance-based classifier (simpan semua data), lazy learner

Tidak membuat sebuah hipotesis, jadinya prediksi data dengan menemukan "grup" yang paling mirip.

Idenya

1. Mengukur jarak dari data masukan ke data di dataset
2. Cari k data yang paling mirip
3. Cari di k data tersebut, mayoritas kelasnya apa



- + Aproksimasi tidak kompleks buat target function yang kompleks
- Costnya high
- Harus consider semua fitur

Naive Bayes

Supervised learning, probabilistic classifier, jadi hipotesisnya akan berbentuk model probabilitas $P(vj)$ dan $P(ai|vj)$

Prediksinya itu menggunakan probabilitas yang paling besar dari $P(vj | a1, a2, \dots, an)$ (jadi yang most likely labelnya untuk kombinasi data seperti itu)

Idenya (training)

1. Ukur semua frekuensi nilai dari setiap atribut
2. Tentuin probabilitas nilai tersebut untuk atribut tersebut, berdasarkan labelnya
3. Cari probabilitas $P(vj)$
4. Hipotesis sebagai sebuah model probabilitas

Idenya (classification)

1. Hitung semua probabilitas dari proyeksi setiap atribut untuk data
2. Kaliin semua untuk dapat probabilitas kelas/label tersebut given set of data untuk setiap atributnya
3. Ambil yang maksimal

05

Example: Play Tennis Dataset

outlook	temp.	humidity	windy	play	outlook	temp.	humidity	windy	play
sunny	hot	high	false	no	sunny	mild	high	false	no
sunny	hot	high	true	no	sunny	cool	normal	false	yes
overcast	hot	high	false	yes	rainy	mild	normal	false	yes
rainy	mild	high	false	yes	sunny	mild	normal	true	yes
rainy	cool	normal	false	yes	overcast	mild	high	true	yes
rainy	cool	normal	true	no	overcast	hot	normal	false	yes
overcast	cool	normal	true	yes	rainy	mild	high	true	no

Frequency of
(sunny|yes) →

outlook	temperature	humidity	windy	play
yes	no	yes	no	yes
sunny	2 3	hot 2 2	high 3 4	false 6 2 9 5
overcast	4 0	mild 4 2	normal 6 1	true 3 3
rainy	3 2	cool 3 1		

Frequency of
(sunny|no) →

Frequency of
class 'yes' → 9

Frequency of
class 'no' → 5

06

Example: Play Tennis Dataset

outlook	temperature	humidity	windy	play
yes	no	yes	no	yes no
sunny	2 3	hot 2 2	high 3 4	false 6 2 9 5
overcast	4 0	mild 4 2	normal 6 1	true 3 3
rainy	3 2	cool 3 1		

Frequency

$P(a_i | v_j)$

outlook	temperature	humidity	windy
yes no	yes no	yes no	yes no
sunny 2/9 3/5	hot 2/9 2/5	high 3/9 4/5	false 6/9 2/5
overcast 4/9 0/5	mild 4/9 2/5	normal 6/9 1/5	true 3/9 3/5
rainy 3/9 2/5	cool 3/9 1/5		

$P(v_j)$

play
yes no 9/14 5/14

Probability Model

EDUNEX ITB

07

Classify New Instance: <Sunny, Cool, High, True>

$P(a_i | v_j)$

outlook	temperature	humidity	windy
yes no	yes no	yes no	yes no
sunny 2/9 3/5	hot 2/9 2/5	high 3/9 4/5	false 6/9 2/5
overcast 4/9 0/5	mild 4/9 2/5	normal 6/9 1/5	true 3/9 3/5

$P(v_j)$

play
yes no 9/14 5/14

$$P(v_j | a_1, a_2, \dots, a_n) = P(v_j) \cdot \prod_i P(a_i | v_j)$$

$P(\text{yes} | \text{sunny, cool, high, true})$

$$= P(\text{yes}) \cdot P(\text{sunny} | \text{yes}) \cdot P(\text{cool} | \text{yes}) \cdot P(\text{high} | \text{yes}) \cdot P(\text{true} | \text{yes}) \\ = 9/14 \cdot 2/9 \cdot 3/9 \cdot 3/9 \cdot 3/9 = 0.0053$$

$P(\text{no} | \text{sunny, cool, high, true})$

$$= P(\text{no}) \cdot P(\text{sunny} | \text{no}) \cdot P(\text{cool} | \text{no}) \cdot P(\text{high} | \text{no}) \cdot P(\text{true} | \text{no}) \\ = 5/14 \cdot 3/5 \cdot 1/5 \cdot 4/5 \cdot 3/5 = 0.0206$$

EDUNEX ITB

Prediction Measurement

Kalau supervised learning itu bisa kita cek hasil prediksinya

		Prediction	
		True	False
Reality	True	Tp True-positive	Fn False-negative
	False	Fp False-positive	Tn True-negative

Depannya itu **benar atau tidaknya**, belakangnya itu **prediksinya**

Instance	Correct Class	Prediction	
1	+	+	TP
2	-	-	TN
3	-	+	FP
4	+	-	FN
5	-	+	FP
6	+	-	FN
7	+	+	TP
8	-	-	TN
9	-	-	TN
10	+	+	TP

Accuracy

Semua yang benar dibagi semua prediksi

		Prediction	
		True	False
Reality	True	Tp True-positive	Fn False-negative
	False	Fp False-positive	Tn True-negative

Fraction of all correct predictions over all predicted instances

$$\text{Accuracy} = \frac{Tp + Tn}{Tp + Fp + Tn + Fn}$$

Precision

Dari semua prediksi benar, berapa yang emang beneran benar?

		Prediction	
		True	False
Reality	True	T _p True-positive	F _n False-negative
	False	F _p False-positive	T _n True-negative

Fraction of positive predictions that are correct

$$\text{Precision} = \frac{T_p}{T_p + F_p}$$

EDUNE)

Recall

Dari yang beneran benar, berapa yang ke prediksi?

		Prediction	
		True	False
Reality	True	T _p True-positive	F _n False-negative
	False	F _p False-positive	T _n True-negative

Fraction of positive instances that are correctly predicted
(retrieved/caught)

$$\text{Recall} = \frac{T_p}{T_p + F_n}$$

EDUNE)

Unsupervised

Unsupervised Learning itu training data nya juga ga punya label, jadinya di cluster
Setiap cluster akan punya centroid (mean M_i)
Buat klasifikasi, masukin ke grup dengan centroid terdekat

K-Means

1. Initialization

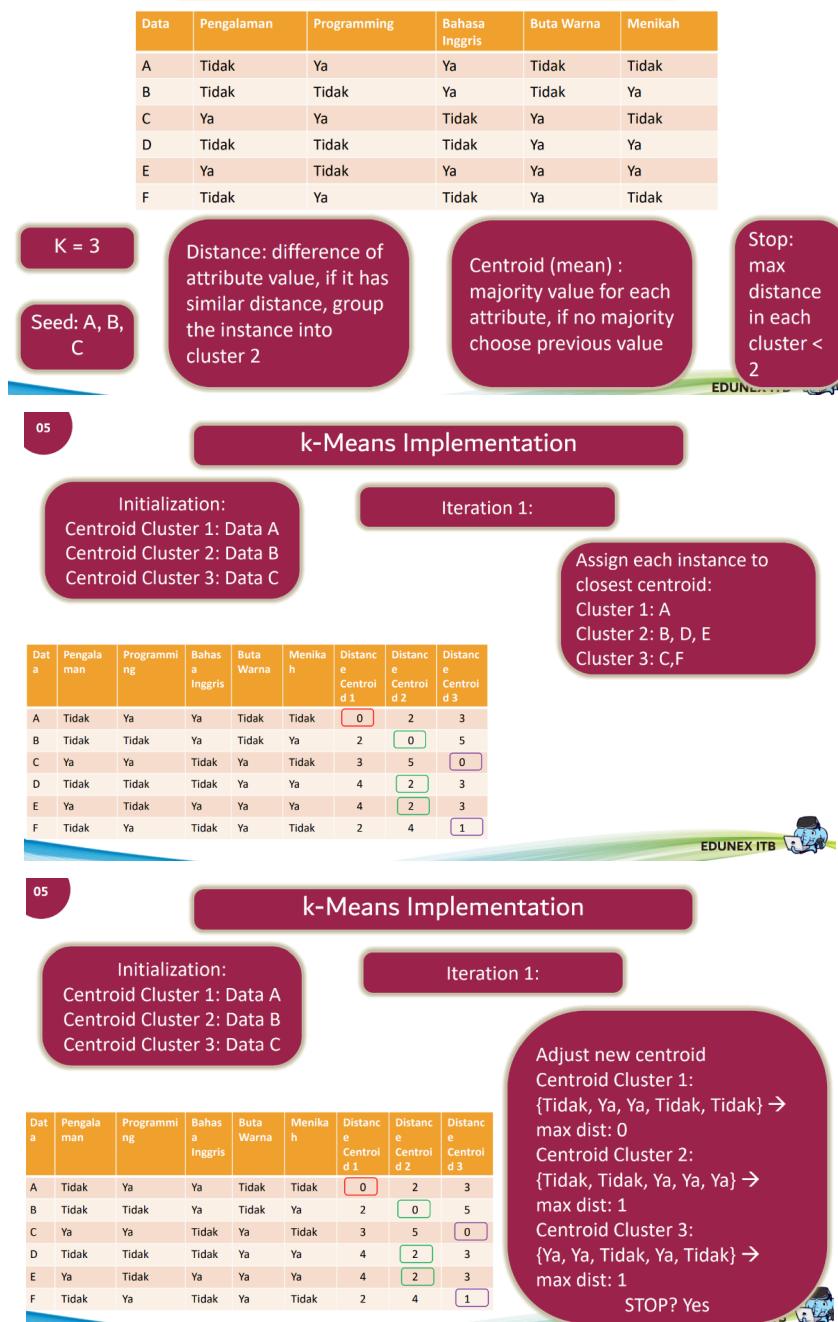
Secara random pilih k seeds buat k cluster

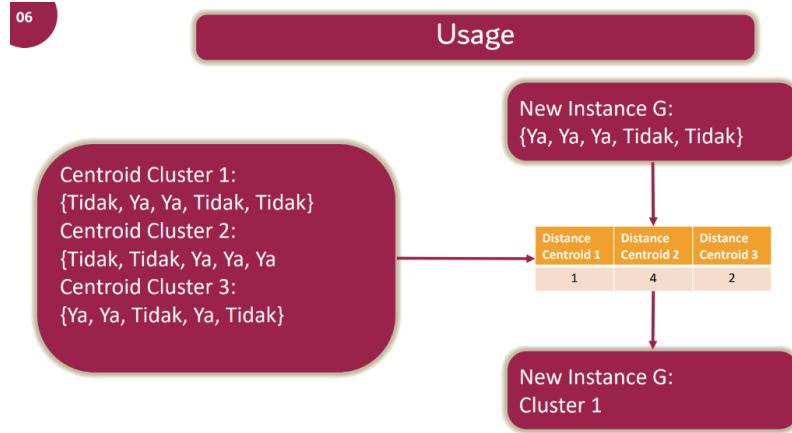
2. Iteration

Assign semua instance dari training data ke centroid terdekat, bikin k cluster

Bikin centroid baru (M_i) untuk setiap cluster (bisa aja data yang gaada)

Stop saat converge/threshold/jumlah iterasi





K-Means Evaluation

Gimana cari K terbaik?

WCSS (Within-Cluster-Sum-of-Squares)

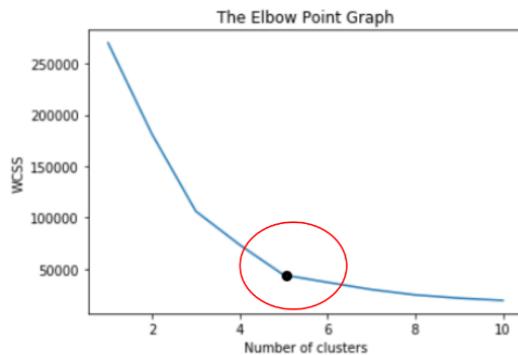
$$WCSS = \sum_{k=1}^n \sum_{i=1}^m ||d_i - C_k||^2$$

Ekspektasinya WCSS kecil, kalau mau 0 ya N instance bikin N cluster, tapi kan bodoh ya

Elbow method

Coba k dari terkecil sampai terbesar, terus cari “elbow” nya

Elbow itu titik dimana penurunannya jadi melambat



Clustering untuk Data Berlabel

Purity

0 bad cluster, 1 good cluster

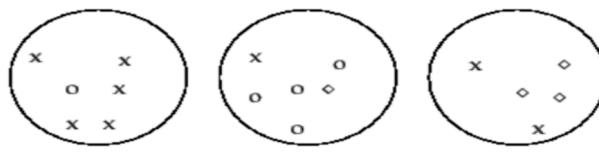
Purity in a cluster:

$$P(C_r) = \frac{1}{n_r} \max_i n_r^i$$

Purity of all clustering result:

$$\begin{aligned} Purity(C) &= \sum_{r=1}^k \frac{n_r}{n} P(C_r) \\ \text{or } &= \frac{1}{n} \sum_{r=1}^k \max_i (n_r^i) \end{aligned}$$

Example Purity Measurement



$$P(C_1) = 5/6$$

$$P(C_2) = 4/6$$

$$P(C_3) = 3/5$$

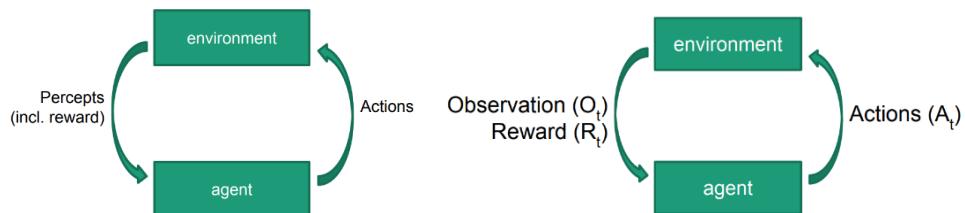
$$P(C) = (5+4+3) / 17 \approx 0.71$$

Intinya label yang mayoritas jadi yang benar untuk cluster tersebut, terus dilihat dia seberapa pure majoritasnya.

Buat cek semua kualitasnya itu n mayoritas setiap cluster dibagi n data

Reinforcement

Carrot and stick method, jadi dia dikasih feedback, bisa reward atau punishment



- Gaada supervisor yang ngasih perfect feedback, dia cuma bisa tau how well it is performing at any step. Jadi dia mau memaksimalkan kinerja itu.
- Feedback itu bakal delayed
- Time itu matters karena kita mentingin sequence
- Aksi dari agen juga mempengaruhi data yang dia lihat

History

Sequence-nya, observation, reward, action (ORA)

$H_t = O_1, R_1, A_1, \dots, A_{t-1}, O_t, R_t$

Bakal mempengaruhi Action yang diambil oleh agen, dan Observation/Reward yang diberikan environment

State

Informasi yang digunakan untuk menentukan apa yang terjadi berikutnya

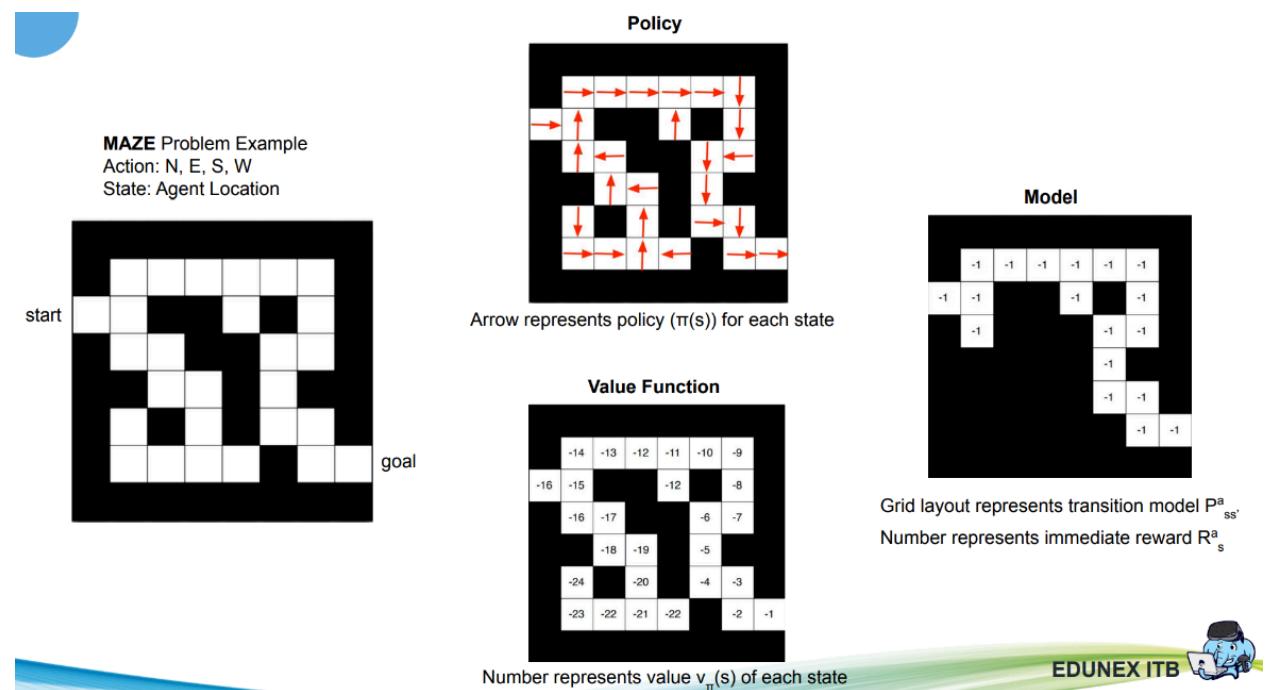
$S_t = f(H_t)$

Sa itu state agent (yang akan mengambil action)

Se itu state environment (data yang digunakan environment untuk memberikan observation dan reward)

Komponen

1. Policy
 - Fungsi behavior agen
 - State → Action
 - Bisa deterministic ($a=\pi(s)$) atau stochastic ($\pi(a|s)=P(A_t=a|S_t=s)$)
2. Value function
 - Evaluasi seberapa bagus state atau action
 - Prediksi dari future reward
 - Bisa digunakan untuk memilih aksi ($v_\pi(s)=E_\pi[R_{t+1}+\gamma R_{t+2}+\dots|S_t=s]$)
3. Model
 - Representasi agen dari environment
 - P prediksi next state
 - ($P_{ss'}^a = P[S_{t+1}=s'|S_t=s, A_t=a]$)
 - R prediksi next immediate reward
 - ($R_s^a = E[R_{t+1}|S_t=s, A_t=a]$)



P itu transition model, ngasih info kepada agen tentang seberapa mungkin berpindah dari state s ke s' setelah mengambil aksi a

R itu reward model, ngasih info ke agen tentang reward yang diharapkan setelah mengambil suatu aksi a di keadaan s

PEAS, P itu kaya aturan main environment, R itu reward yang bakal dikasih sama environment

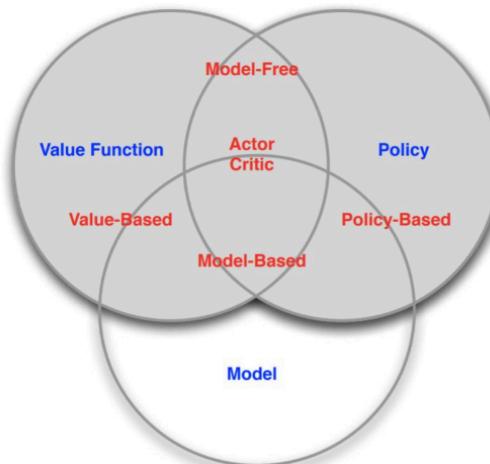
1. Policy $\pi(s)$:

- Persamaan ini menentukan tindakan yang harus diambil agen ketika berada dalam keadaan s . Jika policy deterministik, maka $a = \pi(s)$, yang berarti agen akan selalu mengambil tindakan a ketika berada di keadaan s . Jika policy stokastik, maka $\pi(a|s) = P(A_t = a|S_t = s)$, yang berarti ada probabilitas tertentu P bahwa agen akan mengambil tindakan a ketika berada di keadaan s .

2. Value Function $V(s)$:

- $V_\pi(s) = E[R_{t+1} + \gamma R_{t+2} + \dots | S_t = s]$ adalah ekspektasi (atau rata-rata yang diharapkan) dari jumlah hadiah yang didiskonkan yang akan diperoleh agen dari keadaan s dan seterusnya, mengikuti policy π .
- Di sini γ adalah faktor diskon yang menentukan seberapa penting hadiah di masa depan dibandingkan dengan hadiah segera.

RL Agent Taxonomy



Q-Learning with Temporal Difference

Goal → optimal policy

Menggunakan fungsi Q(s,a)

Asumsi

- Environment fully observable
- Current state dari percept
- Model free

TD Q-learning Method

- $Q(S,A) = Q(S,A) + \alpha[R_s + \gamma \max_a Q(S',a) - Q(S,A)]$
- S: state
- A: selected action
- α : learning rate or step size
- R: reward signal
- γ : discounted factor
- S' : successive state
- a: possible action

```

Q-learning (off-policy TD control) for estimating  $\pi \approx \pi_*$ 

Algorithm parameters: step size  $\alpha \in (0, 1]$ , small  $\varepsilon > 0$ 
Initialize  $Q(s, a)$ , for all  $s \in S^+, a \in \mathcal{A}(s)$ , arbitrarily except that  $Q(\text{terminal}, \cdot) = 0$ 
Loop for each episode:
    Initialize  $S$ 
    Loop for each step of episode:
        Choose  $A$  from  $S$  using policy derived from  $Q$  (e.g.,  $\varepsilon$ -greedy)
        Take action  $A$ , observe  $R, S'$ 
         $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$ 
         $S \leftarrow S'$ 
    until  $S$  is terminal

```

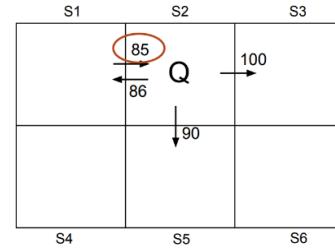
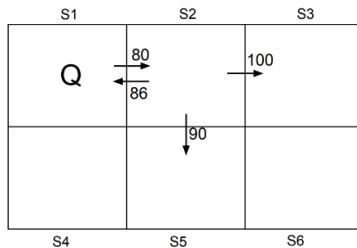
Picture Source: Sutton & Barto, Reinforcement Learning "An Introduction", MIT Press

Example of Q Value Calculation

Reward signal = 10
State: S1, S2, S3, S4, S5, S6
Action: $\uparrow \downarrow \rightarrow \leftarrow$
 $\alpha = 0,5$ dan $\gamma = 0,8$

$$Q(S,A) \leftarrow Q(S,A) + \alpha[R_s + \gamma \max_a Q(S',a) - Q(S,A)]$$

$$Q(S1, \rightarrow) = 80 + 0,5 [10 + 0,8 * \max(86,90,100) - 80] = 85$$



TD with SARSA

Lebih berhati-hati karena update nilai Q dengan aksi yang benar-benar akan diambil berikutnya yang dipilih berdasarkan policy saat ini

SARSA (State-Action-Reward-State-Action)

- $Q(S,A) = Q(S,A) + \alpha[R_s + \gamma Q(S',A') - Q(S,A)]$
- A' : action actually taken in state S'
- $Q(S, A)$ is updated until an action A' is actually taken
- When exploration is happening, SARSA and TD Q-learning are differ
- TD Q-learning: pays no attention to the actual policy being followed
 - off-policy
- SARSA: on-policy

Q-learning itu bakal selalu asumsi kaya agen mengambil step yang paling optimal, tanpa mempertimbangkan policy (cara pemilihan aksi), makanya disebut off-policy
SARSA itu bakal selalu ambil aksi yang emang sesuai sama policy nya

Contoh policy itu e-greedy, dimana agen akan seringnya mengambil Q tertinggi, tapi kadang mengambil acak untuk eksplorasi

Quiz time

Bagaimana SARSA memperlakukan transisi yang mengakibatkan hadiah negatif besar, dan bagaimana hal ini berbeda dengan pendekatan Q-learning?

Di SARSA, karena lebih berhati-hati, maka agen akan lebih menghindari punishment negatif besar karena akan sesuai dengan policy, berbeda dengan Q-learning dimana agen akan selalu memilih Q optimal, yang bisa jadi berujung ke punishment negatif besar. Hal ini terjadi karena SARSA melakukan update berdasarkan tindakan yang memang sudah diambil, sedangkan Q-learning menganggap tindakan optimal yang selalu diambil.

Dalam konteks ϵ -greedy, mengapa penting bagi agen untuk terus melakukan eksplorasi, dan bagaimana ini memengaruhi pembelajaran kebijakan?

Eksplorasi itu penting agar agen tidak stuck di local optima, yang menjadikan hasilnya bias. Eksplorasi juga penting agar agen menemukan hadiah yang mungkin belum ditemukan sebelumnya dan juga membantu membangun pemahaman yang lebih baik tentang environment

Bagaimana faktor diskon γ memengaruhi trade-off antara hadiah jangka pendek dan jangka panjang, dan bagaimana ini bisa mengubah strategi yang dipilih oleh agen dalam pembelajaran?

Alfa itu laju pembelajaran, faktor yang nentuin seberapa besar pengaruh informasi baru terhadap pengetahuan yang sudah dimiliki.

Gamma itu faktor diskon yang menilai pentingnya hadiah di masa depan, kalau gamma makin besar, agen akan makin mempertimbangkan hadiah di masa depan

Jelaskan bagaimana perubahan kecil dalam nilai Q bisa pada akhirnya mempengaruhi kebijakan secara keseluruhan yang dipelajari oleh agen.

Bisa compounding effect, karena nilai Q digunakan untuk memilih tindakan. Bisa mempengaruhi perubahan policy secara signifikan lama-kelamaan. Misal kalau reward nya itu secara konsisten ngasih hasil yang sedikit lebih baik, nilai Q nya akan naik terus, yang menyebabkan agen akan melakukan terus. Menunjukkan gimana RL itu pembelajarannya dinamis dan beradaptasi berdasarkan feedback lingkungan.

Misal si kotak kuning itu punya pilihan action, ngesot, jalan, lompat. Kalau ngesot bagus terus hasilnya, nanti dia jadi ngesot terus.

Apa sih nilai Q?

Nilai Q itu nilai kualitas, prediksi seberapa baik atau berkualitas suatu tindakan yang diambil dari suatu state tertentu, dalam konteks total hadiah yang diharapkan bisa diperoleh di masa depan.

Nilai Q muncul dari interaksi agen dengan lingkungan, kalau misal dia dapet reward positif, nanti dicatet sama agen dalam bentuk nilai Q.

Nilai Q itu gabungin reward saat ini dengan estimasi reward di masa depan, yang didiskon oleh faktor gamma. Jadi Q itu bisa mencerminkan total hadiah yang diharapkan, bukan hadiah langsung.

Jadi State itu juga kaya keadaan si agen nya, misalnya state nya lagi

Q-learning dijamin akan konvergen, itu karena selalu mengasumsikan pemilihan optimal, jadinya selalu akan konvergen ke policy optimal

Inisialisasi nilai Q bisa netral (0), bisa optimis (+, jadi agen lebih mau eksplorasi bagian situ dulu), bisa pesimis (-, jadi agen engga eksplor bagian situ)

Bab 6: Probabilistic Reasoning System

Introduction

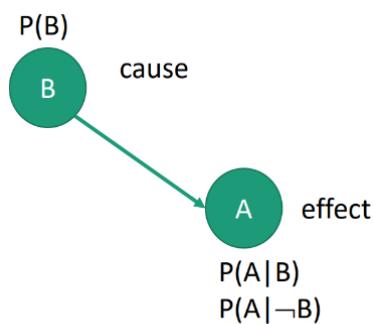
Supervised learning + uncertainty (non deterministic), jadi menggunakan teori probabilitas. Probabilitas disini itu degree of belief, seberapa yakin kita itu akan terjadi.

Jika diberikan suatu state e, apa probabilitas x terjadi? $P(x|e)$

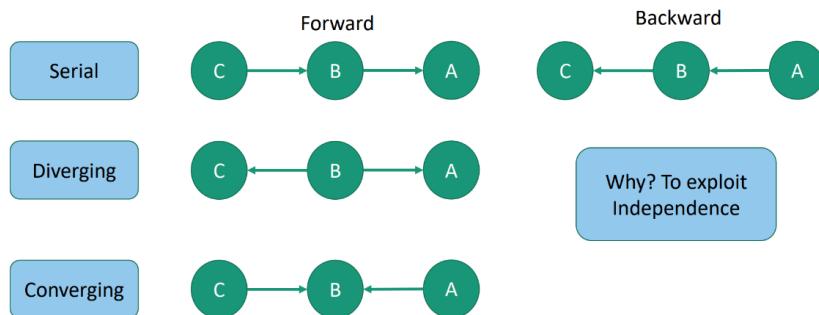
Bisa joint probability distribution, bisa bayesian/belief network

Structure of Bayesian Network

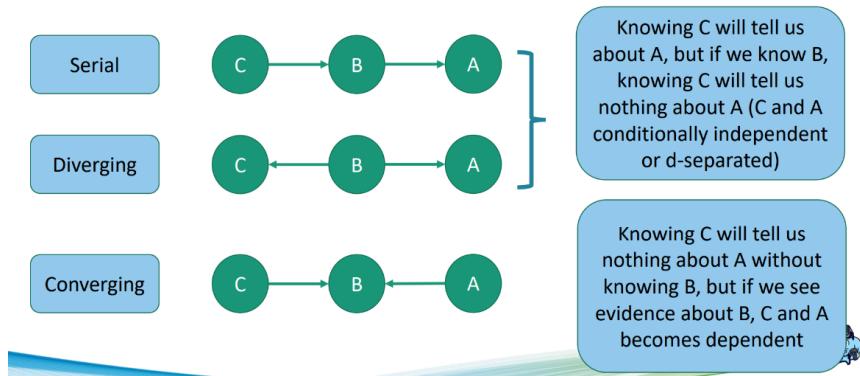
Nodes (variable) & directed arc, membentuk sebuah directed acyclic graph (DAG)



Tipe Koneksi

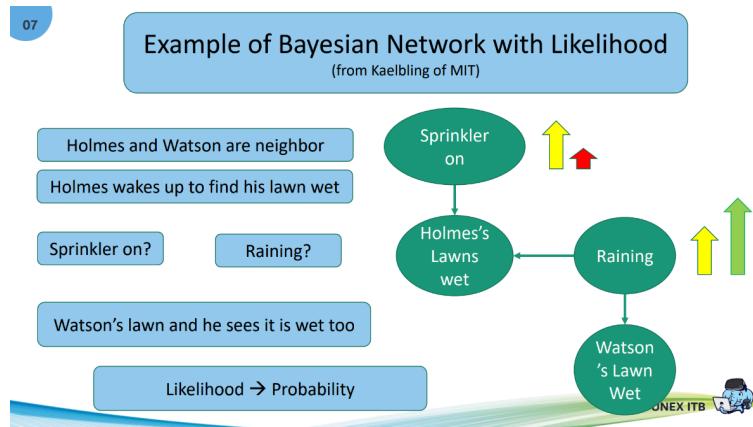


Independence in Connection



Serial and diverging, jika kita tau C, kita bisa tau tentang A. Tapi kalau kita tau B, C jadi ga ngasih tau apa apa tentang A (C & A - conditionally independent, d-separated)

Converging, jika kita tau C tapi ga tau B, ga tau tentang A. Tapi kalau kita tau B, C dan A menjadi dependen



What and Why

Representasi causal dependencies.

BN bisa probabilistic reasoning seperti full joint distribution (full-joint distribution bisa jadi sangat besar)

BN itu independence sama conditional independence relationship bisa sangat ngurangi number of probabilities

lightness	width	category	Prob
1.5	14.6	salmon	0.3
...
8.3	15	Sea bass	0.25
...

In full joint probability distribution, each combination of variable values has information how probable it is.



Components

1. Structure
 - Node (variables)
 - Directed arcs (acyclic graph)
2. Numerical parameters
 - Probability conditional tables

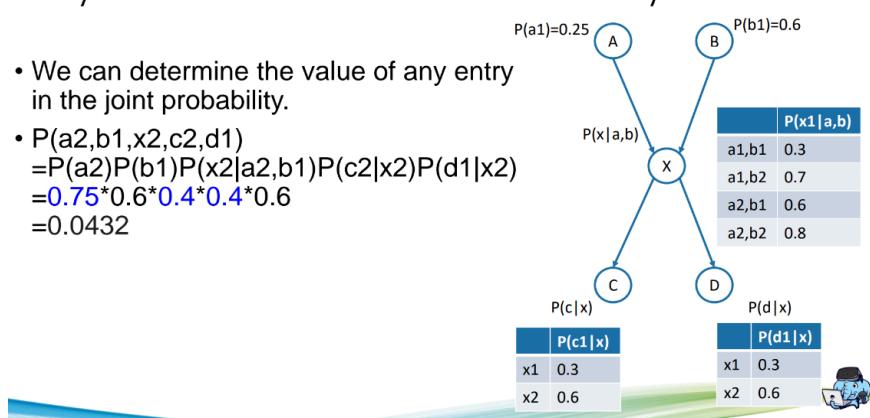
Reduction Number of Probabilities

Kalau full joint probability, 2^N

Kalau bayesian, misal 5, cuma perlu $2 + 2 + 8 + 4 + 4 = 20$ (atau 10 dengan komplement)

Bayesian Network as Joint Probability

- We can determine the value of any entry in the joint probability.
- $P(a_2, b_1, x_2, c_2, d_1)$
 $= P(a_2)P(b_1)P(x_2|a_2, b_1)P(c_2|x_2)P(d_1|x_2)$
 $= \textcolor{blue}{0.75} * 0.6 * \textcolor{blue}{0.4} * \textcolor{blue}{0.4} * 0.6$
 $= 0.0432$

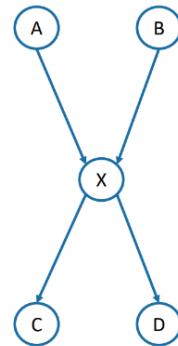


Intinya jadi bisa digabung-gabungin, daripada setiap kombinasi punya 1 probability

Classification in Bayesian Network

Belief Network from Human Expert

- X represents the fish : $x_1 = \text{salmon}$ and $x_2 = \text{sea bass}$.
- X is influenced by A and B.
- A represents time of year: $a_1 = \text{winter}$, $a_2 = \text{spring}$, $a_3 = \text{summer}$ and $a_4 = \text{autumn}$. Probability distribution on A is uniform.
- B represents geographical area where the fish was caught: $b_1 = \text{north Atlantic}$ and $b_2 = \text{south Atlantic}$. The probabilities that any fish came from those areas are 0.6 and 0.4.
- C represents lightness with $c_1 = \text{light}$, $c_2 = \text{medium}$ and $c_3 = \text{dark}$
- D represents thickness with $d_1 = \text{wide}$ and $d_2 = \text{thin}$.



Inference in Bayesian Network

The probability that the fish was caught in the summer in the north Atlantic and is a sea bass that is dark and thin.



The probability that the fish was caught in the summer (a3) in the north Atlantic (b1) and is a sea bass (x2) that is dark (c3) and thin (d2).

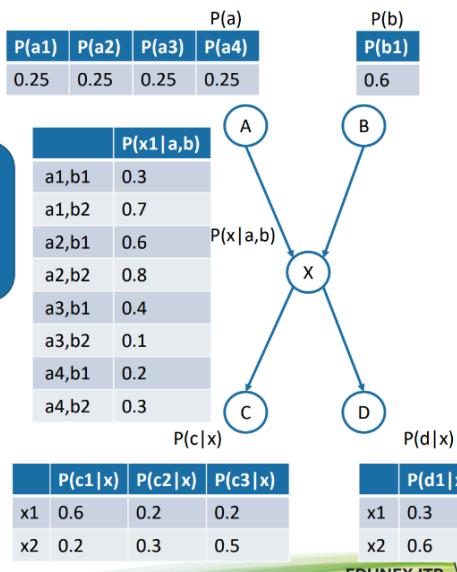


$P(a_3, b_1, x_2, c_3, d_2)$

EDUNEX ITB

Inference: Example

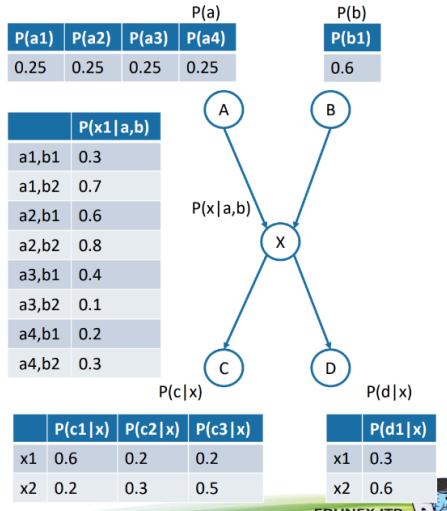
$$\begin{aligned}
 P(a_3, b_1, x_2, c_3, d_2) &= P(a_3)P(b_1)P(x_2|a_3, b_1)P(c_3|x_2)P(d_2|x_2) \\
 &= 0.25 \times 0.6 \times 0.6 \times 0.5 \times 0.4 \\
 &= 0.018
 \end{aligned}$$



Classification

Classify the fish that is light (c_1) and caught in the south Atlantic (b_2), but we do not know what time of year the fish was caught nor its thickness.

Maximum a posterior probability:
 $P(x_1|c_1, b_2)$ vs $P(x_2|c_1, b_2)$



Classification (2)

Q: query

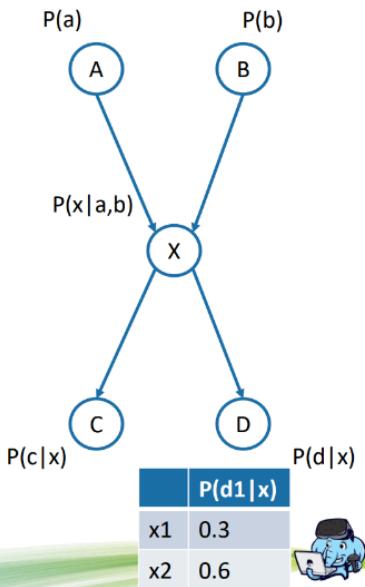
e: evidence of all variables

$$P(Q|e) = P(q,e)/P(e) = \alpha P(Q,e)$$

$$\begin{aligned} P(x_1|c_1, b_2) &= P(x_1, c_1, b_2)/P(c_1, b_2) \\ &= \alpha \sum P(x_1, a, b_2, c_1, d) \\ &= \alpha \sum P(a).P(b_2).P(x_1|a, b_2).P(c_1|x_1).P(d|x_1) \\ &= \alpha P(b_2).P(c_1|x_1) \sum P(a).P(x_1|a, b_2).P(d|x_1) \\ &= \alpha P(b_2).P(c_1|x_1) [\sum P(a).P(x_1|a, b_2)][\sum P(d|x_1)] = \alpha 0.114 \end{aligned}$$

$$\begin{aligned} &P(a_1).P(x_1|a_1, b_2) + \\ &P(a_2).P(x_1|a_2, b_2) + \\ &P(a_3).P(x_1|a_3, b_2) + \\ &P(a_4).P(x_1|a_4, b_2) \end{aligned}$$

$$P(d_1|x_1) + P(d_2|x_1) = 1.0$$



Karena a sama d gatau, jadi jumlahin semua

Karena kalau ga d1 ya d2 buat x manapun (true or false, $d_2 = 1 - P(d_1)$) maka kalau dijumlahin pasti 1

Classification (3)

$$\begin{aligned} P(x_1|c_1, b_2) &= P(x_1, c_1, b_2)/P(c_1, b_2) \\ &= \alpha P(b_2).P(c_1|x_1).[\sum P(a).P(x_1|a, b_2)]. \\ &[\sum P(d|x_1)] = \alpha 0.114 \end{aligned}$$

$$\begin{aligned} P(x_2|c_1, b_2) &= P(x_2, c_1, b_2)/P(c_1, b_2) \\ &= \alpha P(b_2).P(c_1|x_2) [\sum P(a)]. \\ &P(x_2|a, b_2)][\sum P(d|x_2)] = \alpha 0.042 \end{aligned}$$

Normalize:

$$P(x_1|c_1, b_2) = 0.73$$

$$P(x_2|c_1, b_2) = 0.27$$

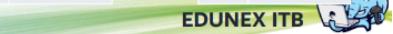
Decision: $x_1 = \text{salmon}$

	P(a1)	P(a2)	P(a3)	P(a4)	P(a)
	0.25	0.25	0.25	0.25	0.6

	P(x1 a,b)
a1,b1	0.3
a1,b2	0.7
a2,b1	0.6
a2,b2	0.8
a3,b1	0.4
a3,b2	0.1
a4,b1	0.2
a4,b2	0.3

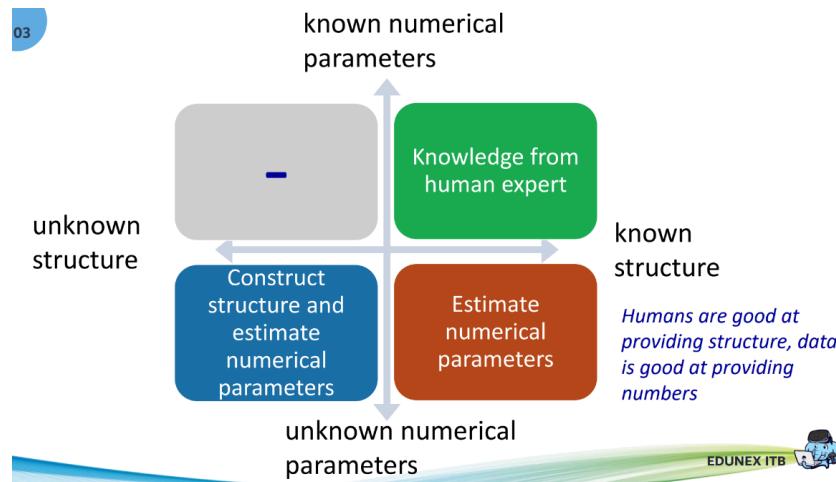
	P(c x)	P(d x)
x1	0.6	0.3
x2	0.2	0.6

	P(d1 x)
x1	0.3
x2	0.6



Learning from Data

Masalahnya kalau BN dibikin sama human itu susah, lama, malah jadi bottleneck, dan juga ga reliable probabilitynya



Struktur lebih bagus sama human, numbers lebih bagus sama data

04

Parameter Estimation

Given structure
with m nodes

Given a data set

$$D = \{<v_1^1, \dots, v_m^1>, \dots, <v_1^k, \dots, v_m^k>\}$$

Count $\#(V_i=T)$, $\#(V_i=F)$,
 $\#(V_i=T, V_j=T)$, $\#(V_i=T, V_j=F)$

Variable V_i with no parent
 $P(V_i) \approx \frac{\#(V_i = T)}{k}$

Variable V_i with parent V_j
 $P(V_i|V_j) \approx \frac{\#(V_i = T, V_j = T)}{\#(V_j = T)}$
 $P(V_i|\neg V_j) \approx \frac{\#(V_i = T, V_j = F)}{\#(V_j = F)}$

EDUNEX ITB 

Smoothing: Avoid Probability = 0

Variable V_i with no parent
 $P(V_i) \approx \frac{\#(V_i = T) + 1}{k + 2}$

Variable V_i with parent V_j
 $P(V_i|V_j) \approx \frac{\#(V_i = T, V_j = T) + 1}{\#(V_j = T) + 2}$
 $P(V_i|\neg V_j) \approx \frac{\#(V_i = T, V_j = F) + 1}{\#(V_j = F) + 2}$

+2 di V_i atas itu karena V_i itu binary variable, (soalnya kan buat $V_i = T$ dia +1 dan $V_i = F$ juga +1), jadinya dia buat compensate perlu +2. Kalau dia n-valued, ditambah n si pembaginya.

Construct BN Structure

1. Determine set of variable
2. Determine ordering of variables $X_1..X_m$
3. Loop: add X_i , terus tentuin parent dari X_i dari $X_1..X_{i-1}$

Construct BN Structure: Example

Based on causal knowledge:

- Set of variables: A,B,C,D,X
- Ordering of variables: A,B,C,D,X
- Add A: no parent.
- Add B: is A parent of B ?
- Add C: parent of C ?
- Add D: parent of D ?
- Add X: parent of X ?

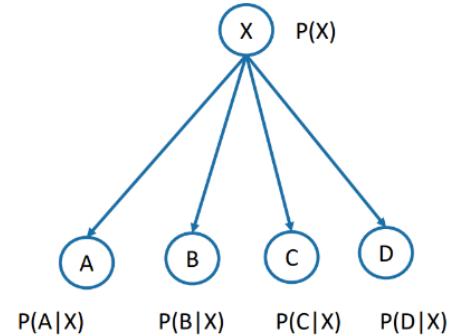
Based on data:

- Set of variables: A,B,C,D,X
- Ordering of variables: A,B,C,D,X
- Add A: no parent.
- Add B: $P(B|A)=P(B)$?
- Add C:
 - $P(C|A)=P(C)$?
 - $P(C|B)=P(C)$?
 - $P(C|A,B)=P(C|A)$?
 - $P(C|A,B)=P(C|B)$?
 - $P(C|A,B)=P(C)$?
- Etc.

Construct BN Structure: Example 2

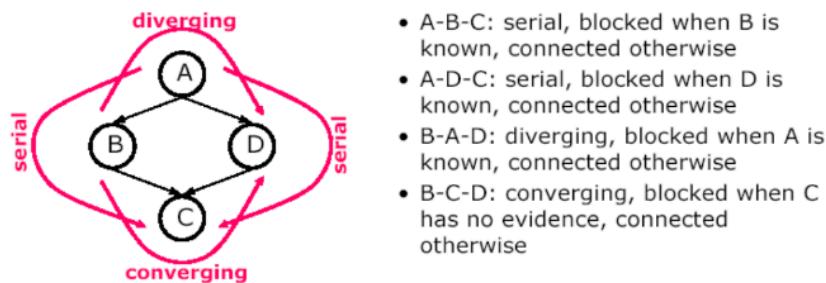
Based on causal knowledge:

- Set of variables: X,A,B,C,D
- Ordering of variables: X,A,B,C,D
- Add X: no parent.
- Add A: $\text{parent}(A)=X$.
- Add B: $\text{parent}(B)=X$. $P(B|A,X)=P(B|X)$
- Add C: $\text{parent}(C)=X$. $P(C|A,B,X)=P(C|X)$
- Add D: $\text{parent}(D)=X$. $P(D|A,B,C,X)=P(D|X)$



D Separation

- Two variables A and B are d-separated iff for every path between them, there is an intermediate variable V such that either
 - The connection is serial or diverging and V is known
 - The connection is converging and neither V nor any descendant is instantiated
 - Two variables are d-connected iff they are not d-separated



D-separated itu kalau ada yang block, dari set variabel observasi.
Jalur aktif kalau info bisa mengalir dari satu ujung ke ujung lainnya

Kalau d-separated, dia gabisa ngasih informasi