TDT4171 - Artificial Intelligence Methods Final Exam Spring 2024

Introduction

- The exam contains 9 tasks with several subtasks in each task.
- Each subtask is worth a certain number of % points, adding up to the number of points specified for the task and 100% for the complete exam.
- If the description of the task is ambiguous, specify your assumptions and solve the task according to these assumptions.
- To get maximum score, the answer should be correct, complete, well-justified and precise. Partial points will be given for answers that are somewhat lacking in content or quality.
- In several tasks, we specified a range for how many sentences should be used for the answer. This is a recommended number of sentences to give you an idea about the expected level of detail. Minor deviations from this range are acceptable. If your answer is significantly longer, consider summarizing the main points in fewer sentences. If your answer is shorter, consider expanding it a bit.

1 [6%] AI foundation

1.1 [3%]

Which of these statements are True or False? +0.5% for correct and -0.5% for incorrect, 0% points for no answer.

1. The main goal of this course is to understand what intelligence is.
2. By definition, Strong AI outperforms Weak AI in all tasks.
3. Weak AI can pass a Turing test.
4. Acting rationally means acting like humans do.
5. An AI agent with faulty sensors can still act rationally.
6. To be rational, an agent must be able to compute utilities and probabilities.

Solution

1. False
2. False
3. True
4. False
5. True
6. False

Grading

Automatic

1.2 [3%]

List any five ethical concerns / misuses of AI.

Solution

1. Lethal autonomous weapons
2. Large scale surveillance and privacy.
3. Fairness and bias
4. Lack of transparency
5. Work automation, people loosing jobs
6. Concentration of wealth

Grading

There can be variations, e.g. more specific concerns. Give % proportional to how many items are mentioned.

2 [10%] Uncertainty and Bayesian networks

2.1 [3%]

Given the following joint probability distribution:

	toothache		¬ toothache	
	catch	¬ catch	catch	¬ catch
cavity	.13	.03	.07	.01
¬ cavity	.01	.06	.12	.57

Compute the following probabilities:

1. $P(toothache)$
2. $P(cavity \vee toothache)$
3. $P(\neg cavity \mid toothache)$

Round your answers to two decimals, e.g. 1.236 should be rounded to 1.24

Solution

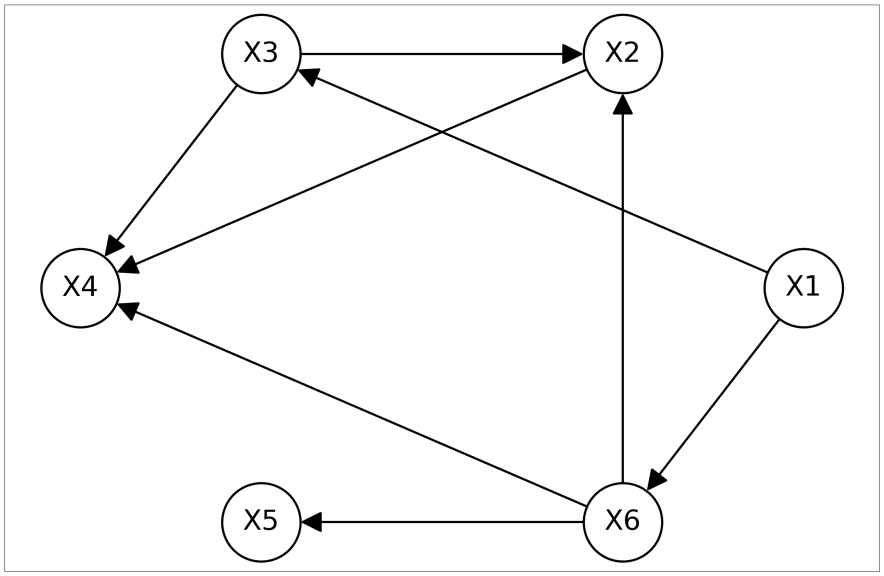
1. $P(toothache) = 0.13 + 0.03 + 0.01 + 0.06 = 0.23$
2. $P(cavity \vee toothache) = 0.13 + 0.03 + 0.01 + 0.06 + 0.07 + 0.01 = 0.31$
3. $P(\neg cavity \mid toothache) = P(\neg cavity \wedge toothache)/P(toothache) = (0.01 + 0.06)/(0.13 + 0.03 + 0.01 + 0.06) = 0.30$

Grading

Automatic

2.2 [**7%**]

Given the following Bayesian network:



Answer these Yes/No questions. +1% for correct and -1% for incorrect, 0% points for no answer. +2% for all correct.

1. Is X1 independent of X4?
2. Is X3 independent of X6 given X2?
3. Is X1 independent of X5 given {X4, X6}?
4. Is X1 independent of X2 given {X4, X5, X6}?
5. Is X3 independent of X5 given {X1, X2, X4, X6}?

Solution

1. No
2. No
3. Yes
4. No
5. Yes

Grading

Automatic

3 [16%] Decision networks

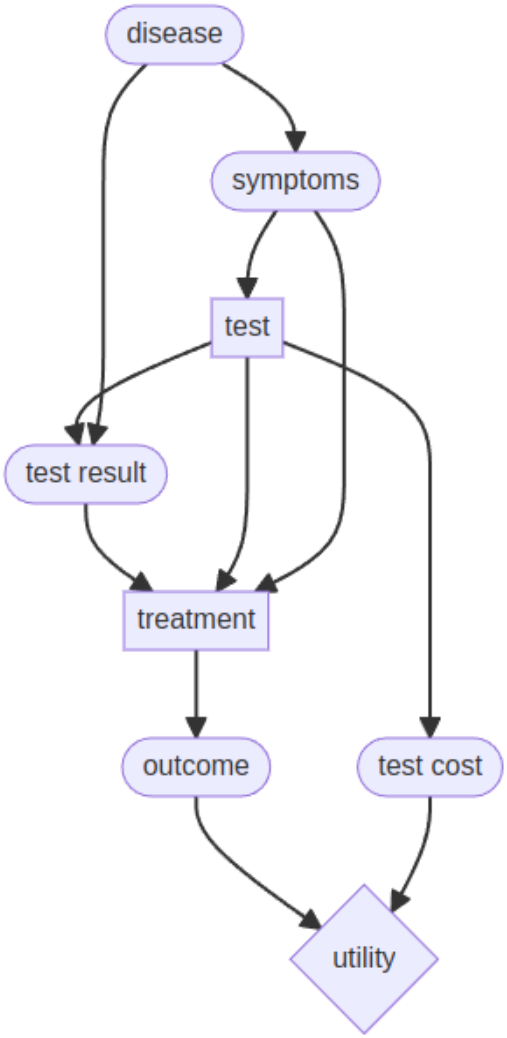
3.1 [5%]

Consider a scenario where a doctor is diagnosing a patient with some symptoms. The doctor needs to make two decisions: (1) what kind of test to perform and (2) what treatment to prescribe.

Construct the decision network for this scenario. Include nodes for disease, symptoms, test, test cost, test results, treatment, treatment outcome and utility. Use ovals for chance nodes, rectangles for decision nodes and diamonds for utility nodes.

Draw the structure (the directed acyclic graph) of this model on **paper**. You are not asked to provide conditional probability tables. However, make your model so that it would be as easy as possible for a domain expert to provide the required probabilities.

Solution



Some other variations are acceptable as well. Mostly looking at the overall understanding of how the decision network can be structured.

Grading

- Full points for all nodes with the correct type and edges with the correct direction. Missing or adding one, two edges is ok with some explanation.
- Partial points if most of the network is correct.
- No points if the student doesn't know what a decision network is.

3.2 [4%]

Given 3.1, specify the conditional probability table for the node representing test result. Possible values for disease: flu and allergy. Possible values for test: flu test and allergy test. Possible values for test results: positive and negative.

Solution

disease	test	test result	probability, %
flu	flu	positive	70
flu	flu	negative	30
flu	allergy	positive	30
flu	allergy	negative	70
allergy	flu	positive	10
allergy	flu	negative	90
allergy	allergy	positive	80
allergy	allergy	negative	20

Alternative without negative test results, i.e. (100 - positive) with the same disease and test values.

disease	test	test result	probability, %
flu	flu	positive	70
flu	allergy	positive	30
allergy	flu	positive	10
allergy	allergy	positive	80

Grading

We are checking that the student understand what conditional probability table is. Probability of positive result for a disease with the corresponding test should be higher than when using another test. Important that rows with the same disease and test values should sum up to 100%. Specific numbers are not important.

3.3 [3%]

Design utility for the network in 3.1 The utility should be based on outcome and test cost. You may use an additive value function. Possible values for the outcome: No symptoms, fewer symptoms, same symptoms. Possible values for the test cost: high, moderate, low.

Solution

Assign utility values to variables.

Outcome: 1. No symptoms: 100 2. Fewer symptoms: 50 3. Same symptoms: 0

Test cost:

1. High: -100
2. Moderate: -50
3. Low: -10

U(outcome, cost) = outcome * 3 + cost

Grading

The students are expected to know how to define simple utility. Concrete numbers don't matter, just needs to be something that makes sense.

3.4 [4%]

Specify the general algorithm (briefly describe steps) for evaluating decision networks, i.e. how they are used for making decisions. You don't need to explain how to calculate posterior probabilities.

Solution

1. Set the evidence variables for the current state.
2. For each possible value of the decision node
    1. Set the decision node to that value.
    2. Calculate the posterior probabilities for the parent nodes of the utility node.
    3. Calculate the resulting utility for the action.
3. Return the action with the highest utility

Grading

- Full points for including all the steps.
- Partial points if some steps are missed.
- No points if the algorithm is mostly wrong.

4 [20%] Probabilistic reasoning over time - Hidden Markov model

One of the challenges with solar energy production in the Nordic countries is snow covering solar panels. Solar panels can still produce some energy if a snow layer is thin or only part of the panel is covered. There are no sensors that can measure it directly, so we have to infer whether the panel is covered based on measured production.

Consider the following probabilities:

- Given that the panel is covered with snow, the probability that a solar panel produces energy is 0.1.
- Given that the panel is **not** covered with snow, the probability that a solar panel produces energy is 0.7.

- Given that the panel is covered with snow during the current hour, the probability that the panel will be covered with snow for the next hour is 0.9.
- Given that the panel is **not** covered with snow during the current hour, the probability that it will **not** be covered with snow for the next hour is 0.8.
- The prior probability for the panel being covered with snow is 0.3.

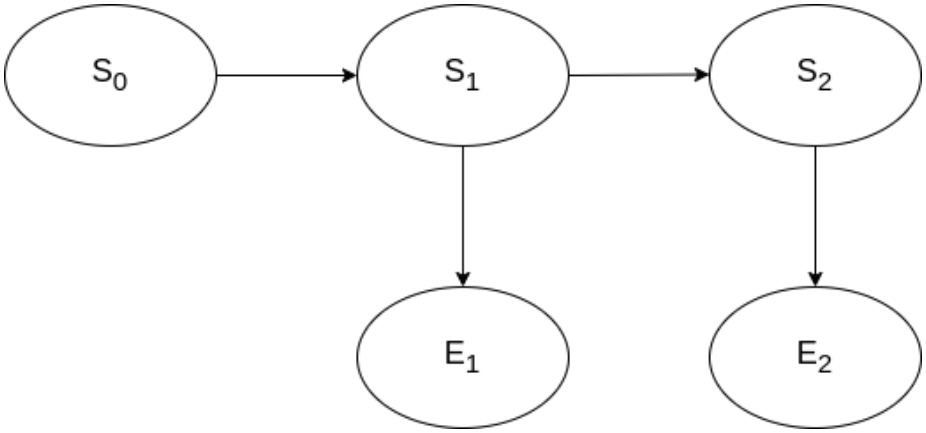
4.1 [8%]

Specify a hidden Markov model representing the problem above. You need to specify the structure and the probability tables. The structure should include steps (hours) 0, 1 and 2. Draw the model and tables on **paper**.

Use the following notation:

- S_t - random variable for the panel covered with snow at step t
- s_t - value for S_t at step t
- E_t - random variable for the panel producing energy at step t
- e_t - value for E_t at step t

Solution



Transition model:

S_{t-1}	$P(S_t \mid S_{t-1})$
t	0.9
f	0.2

or

S_{t-1}	S_t	$P(S_t \mid S_{t-1})$
t	t	0.9
t	f	0.1
f	t	0.2
f	f	0.8

or

S_{t-1}	$P(S_t = s_t \mid S_{t-1})$	$P(S_t = \neg s_t \mid S_{t-1})$
t	0.9	0.1
f	0.2	0.8

Observation model:

S_t	$P(E_t \mid S_t)$
t	0.1
f	0.7

or

S_t	E_t	$P(E_t \mid S_t)$
t	t	0.1
t	f	0.9
f	t	0.7
f	f	0.3

or

S_t	$P(E_t = e_t \mid S_t)$	$P(E_t = \neg e_t \mid S_t)$
t	0.1	0.9
f	0.7	0.3

I have also seen several other variations for students that are considered correct.

4.2 [3%]

Given the hidden Markov model from 4.1, are these statements true or false? +0.5% for correct and -0.5% for incorrect, 0% points for no answer.

1. $P(S_t|E_t)$ is called the observation model.

2. First-order Markov assumption can be expressed as $P(S_t|S_{0:t-1}) = P(S_t|S_{t-1})$
3. Filtering is the task of computing $P(S_k|e_{1:t})$ where t is the most recent time step and $0 \leq k < t$.
4. Prediction is the task of computing $P(S_{t+k}|e_{1:t})$ where t is the most recent time step and $k > 0$.
5. Smoothing is the task of computing $P(S_t|e_{1:t})$ where t is the most recent time step.
6. Most likely explanation is the task of computing $P(S_{1:t}|e_{1:t})$ where t is the most recent time step.

Solution

1. False
2. True
3. False
4. True
5. False
6. False

Grading

Automatic

4.3 [9%]

Compute the probability of the panel being covered with snow at the second hour given that our measurements show energy production at hour 1 and no production at hour 2. Show your computations step by step for steps 0, 1 and 2. Show computations in symbolic form (variables, probabilities and distributions) before replacing them with numbers. In you computations, round intermediate and final results to two decimals.

Solution

These computations almost exactly follow the example at page 485 and lecture 4 but with different symbols and numbers.

Hour 0

No energy measurements available, we only have prior beliefs:

$$P(S_0) = \langle 0.3, 0.7 \rangle$$

Hour 1

We measured some energy production, so $e_1 = t$

Predict from t_0 to t_1:

$$P(S_1) = \sum_{s_0} P(S_1 \mid s_0)p(s_0) = \langle 0.9, 0.1 \rangle \times 0.3 + \langle 0.2, 0.8 \rangle \times 0.7 = \langle 0.41, 0.59 \rangle$$

Update given $e_1 = t$.

$P(S_1 \mid e_1) = \alpha P(e_1 \mid S_1)P(S_1) = \alpha\langle 0.1, 0.7\rangle\langle 0.41, 0.59\rangle = \alpha\langle 0.04, 0.41\rangle = \langle 0.09, 0.91\rangle$

α is a normalization constant.

Hour 2

We measured no energy production, so $e_2 = f$

Predict from t_1 to t_2:

$P(S_2 \mid e_1) = \sum_{s_1} P(S_2 \mid s_1)p(s_1 \mid e_1) = \langle 0.9, 0.1\rangle \times 0.09 + \langle 0.2, 0.8\rangle \times 0.91 = \langle 0.26, 0.74\rangle$

Update given $e_2 = f$:

$P(S_2 \mid e_1, e_2) = \alpha P(e_2 \mid S_2)P(S_2 \mid e_1) = \alpha\langle 0.9, 0.3\rangle\langle 0.26, 0.74\rangle = \alpha\langle 0.23, 0.22\rangle = \langle 0.51, 0.49\rangle$

Grading

- Full points for correct or almost correct computation flow even if the numbers are a bit wrong. Check that the distributions are at least somewhat close to the solution.
- Partial points if the student made major mistakes in the computation in symbolic level.
- No points if the flow is completely incorrect.

5 [10%] Artificial neural networks - Gradient descent

Given the Gradient Descent algorithm for the perceptron:

1. Initialize each w_i to some small random value
2. Until the termination condition is met:
    1. Initialize: $\Delta w_i \leftarrow 0$
    2. For each $\langle \vec{x}, t\rangle$ in D:
        - Input $\vec{x}$ to the unit and compute the output o
        - For each w_i: $\Delta w_i \leftarrow \Delta w_i - \eta \cdot 2(-x_i(t - o))$
    3. For each w_i: $w_i \leftarrow w_i + \Delta w_i$

5.1 [5%]

Explain what the following symbols mean and their role in this algorithm: D, $\vec{x}$, t, w, η. Answer with 1-2 sentences per symbol.

Solution

- D - training data set containing pairs with feature vectors and target values
- $\vec{x}$ - feature vectors, characteristic of a training instance, provided as input to the perceptron.

- t - target value is what we want our perceptron to learn how to predict
- w - weights that are multiplied by inputs, these are adjusted during learning
- η - learning rate, determines how much to adjust weights for each update

5.2 [5%]

Where is the gradient in this algorithm, and why does it make sense to use the gradient this way? Answer with 2-10 sentences.

Solution

- The gradient in this algorithm is represented by the term $\cdot 2(-x_i(t-o))$.
- This term is derived from the partial derivative of the squared error function with respect to each weight w_i.
- Using the gradient this way makes sense because it indicates the direction and magnitude of the steepest ascent in the error function's value.
- By subtracting this gradient term from the current weights moves the weights in the direction that minimizes the error, improving the model's accuracy iteratively.

6 Deep learning [10%]

6.1 [2%]

What is "deep" in deep learning, and why does it help? Answer with 3-10 sentences.

Solution

- "Deep" in deep learning refers to the use of multiple layers of neurons in a neural network, forming a deep architecture as opposed to a shallow one with only a few layers.
- This depth allows the model to learn and represent complex features and patterns at various levels of abstraction.
- Each layer captures higher-level features built upon the lower-level features extracted by previous layers.
- This enables deep learning models to identify patterns that shallow models can't.

6.2 [4%]

What is overfitting, and how can we avoid it in deep learning? Answer with 3-10 sentences.

Solution

Overfitting occurs when a deep learning model learns the training data too well, including its noise and outliers, resulting in poor generalization to new, unseen data. To avoid overfitting, several techniques can be employed: using a larger and more diverse dataset, implementing regularization methods like L1 or L2

regularization, applying dropout to randomly deactivate neurons during training, and utilizing early stopping to halt training when performance on a validation set begins to degrade. Data augmentation and cross-validation can also help in creating a more robust model that generalizes better.

6.3 [4%]

Explain why convolution networks work better for image data than a multilayer perceptron. Answer with 3-10 sentences.

Solution

Unlike MLP, CNNs leverage the spatial structure of images through convolutional layers. These layers apply convolutional filters that slide across the image detecting patterns such as edges, textures, and more complex structures at different locations. The same set of filters is applied across the whole image with shared weights. This reduces the number of parameters, making the model more efficient, faster to learn. CNNs also use pooling layers to reduce the spatial dimensions and amplifying the most prominent signals. This also makes the model more computationally efficient and less prone to overfitting.

7 [8%] Natural language processing

7.1 [4%]

In natural language processing, what are word embeddings, and why are they better for many NLP tasks than one-hot encoding? Answer with 2-5 sentences.

Solution

In natural language processing, word embeddings are dense vector representations that capture the semantic meanings of words by placing similar words closer together in the vector space. They are better than one-hot encoding because they provide meaningful relationships between words and reduce the dimensionality, improving the performance and efficiency of NLP models.

7.2 [4%]

How would you use word embeddings to classify emails as spam, not spam? Here you only need to outline the general idea. Answer with 2-5 sentences.

Solution

1. Convert the words in each email into their corresponding word embedding vectors using a pre-trained embedding model.
2. Aggregate these word embeddings to form a single fixed-size representation for each email, such as by averaging the embeddings.
3. Feed these aggregated embeddings into a classification model, such as neural network, which is trained to distinguish between spam and not spam emails based on these features.

4. Finally, the trained model can classify new emails based on their aggregated word embeddings.

8 [10%] Reinforcement learning

8.1 [2%]

What does "reinforcement" stand for in reinforcement learning? How is it different from supervised and unsupervised learning? Answer with 2-5 sentences.

Solution

Reinforcement refers to the occasional rewards the agent receives from the environment based on the actions taken. This reinforces behaviors that lead to higher rewards, guiding the agent to learn optimal strategies over time. Unlike supervised learning, which relies on labeled data, and unsupervised learning, which finds patterns in data without labels, reinforcement learning focuses on interaction with an environment to maximize cumulative rewards over multiple steps, making it particularly suitable for sequential decision-making tasks.

8.2 [4%]

How is reinforcement learning (RL) different from solving a Markov decision process (MDP)? Answer with 2-5 sentences.

Solution

- RL is used when the agent must learn the optimal policy through interaction with an environment. In contrast, we can find an optimal policy for MDP without interacting with an environment.
- In RL the agent doesn't know the transition probabilities and reward functions a priori. In contrast, when solving an MDP these dynamics are fully known.
- RL relies on exploration and exploitation to learn the optimal actions. In contrast, solving an MDP involves no exploration.

8.3 [4%]

Explain how deep learning is combined with reinforcement learning to play Flappy Bird. If you don't know Flappy Bird use Mario or some other simple arcade game as an example. You don't need to explain the complete architecture or details of the learning algorithm. Focus on how deep learning is integrated into reinforcement learning. Answer with 3-10 sentences.

Solution

For Flappy Bird, we can use Deep Q-Learning.

- Deep neural network is used to approximate the Q-function.

- Q-function estimates the expected rewards for taking certain actions in given states.
- The game's current state is represented by pixel colors in several consequent game screenshot.
- The state is fed into a convolutional neural network that outputs a vector of values containing q-values for each action.
- During testing we select an action with highest q-value.
- Training is similar to standard q-learning, but q-function is learned by updating weights in a deep neural network.

9 [10%] Case-based reasoning

9.1 [5%]

Explain each step in CBR cycle. 1-2 sentences per step.

Solution

1. Retrieve: the most similar case or cases: The case(s) with the most similar problem description(s)
2. Reuse: the information/experience stored in the solution descriptions of the retrieved case(s) to solve the presented problem
3. Revise: the retrieved solution if it is necessary to solve the presented problem in a satisfying way.
4. Retain: the tested adapted new solution/experience as a new case, consisting of the presented problem description and the adapted solution description as a new experience in the case base

9.2 [5%]

Find a problem/application where case-based reasoning would perform better than deep learning? Explain why it would be better for this problem. Answer with 3-10 sentences.

Solution

- An example CBR would perform better than deep learning in medical diagnosis for rare diseases.
- In this scenario, there may be limited data available for deep learning models to effectively learn and generalize patterns.
- CBR can directly utilize the specific details and nuances of past cases without requiring a large amount of training data.
- CBR can adapt solutions of previous similar cases to a new one, accounting for differences between new and previous cases.
- Each new case can significantly improve the accuracy of a CBR system, which is important for rare diseases.
- Reasoning in CBR is much easier to explain, verify and correct manually, which makes for a better decision-support system.

- CBR doesn't require computation resources for training large models as in deep learning.