Final Review

Chen-Yu Wei

Logistics

- Time: 2PM-4PM, December 17
- Location: Olsson Hall 009 (usual classroom)
- Open notes
  - Just paper-based notes/notebooks, no electronic devices
  - No textbook
- Let me know earlier if you need any accommodation

Topics

- Bayesian Network (focusing on the content after Page 28 in the BN slides, though it’s unavoidable that it will be related to some previous content)
- (Hidden) Markov Model
- Machine Learning
- Deep Learning and Applications
- Reinforcement Learning

Like midterm, the final exam is not going to be “easy”. But we’ll try to include more basic questions.

Bayesian Network

Learning Objective 1

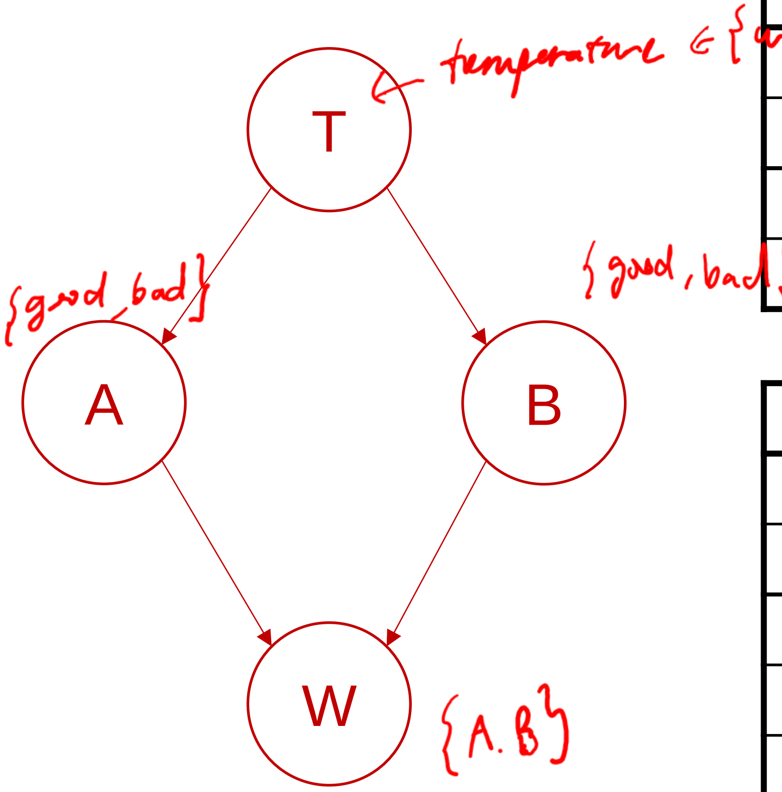
- Formulate a problem as a Bayesian network

Learning Objective 1 – Example

- Two teams A and B are competing in a baseball match.
- If the weather is **warm**, then
  - A is in a good condition with probability 0.6
  - B is in a good condition with probability 0.8
- If the weather is **cold**, then
  - A is in a good condition with probability 0.5
  - B is in a good condition with probability 0.3
- If both team are in the same condition, B wins with probability 0.6
- Otherwise, the good-conditioned team wins with probability 0.8
- The weather is warm with probability 0.4 and cold with probability 0.6

Learning Objective 1 – Example

T	P(T)
Warm	0.4
Cold	0.6



T	A	P(A|T)
Warm	Good	0.6
Warm	Bad	0.4
Cold	Good	0.5
Cold	Bad	0.5

T	B	P(B|T)
Warm	Good	0.8
Warm	Bad	0.2
Cold	Good	0.3
Cold	Bad	0.7

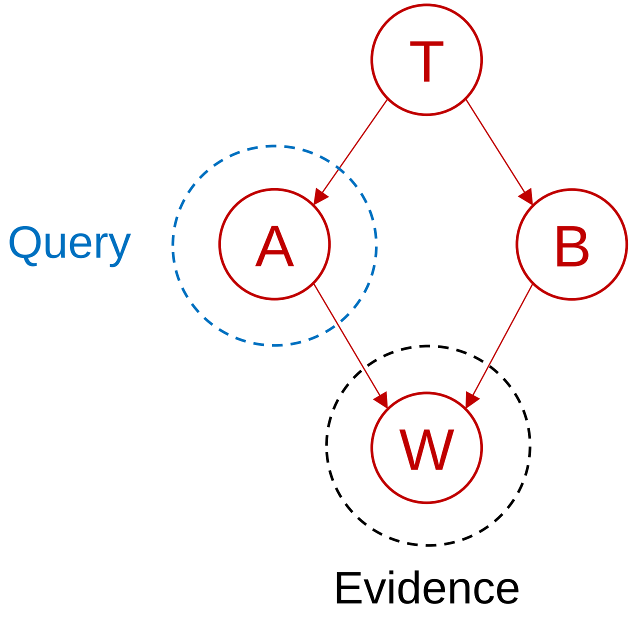
A	B	W	P(W|A,B)
Good	Good	A	0.4
Good	Good	B	0.6
Good	Bad	A	0.8
Good	Bad	B	0.2
Bad	Good	A	0.2
Bad	Good	B	0.8
Bad	Bad	A	0.4
Bad	Bad	B	0.6

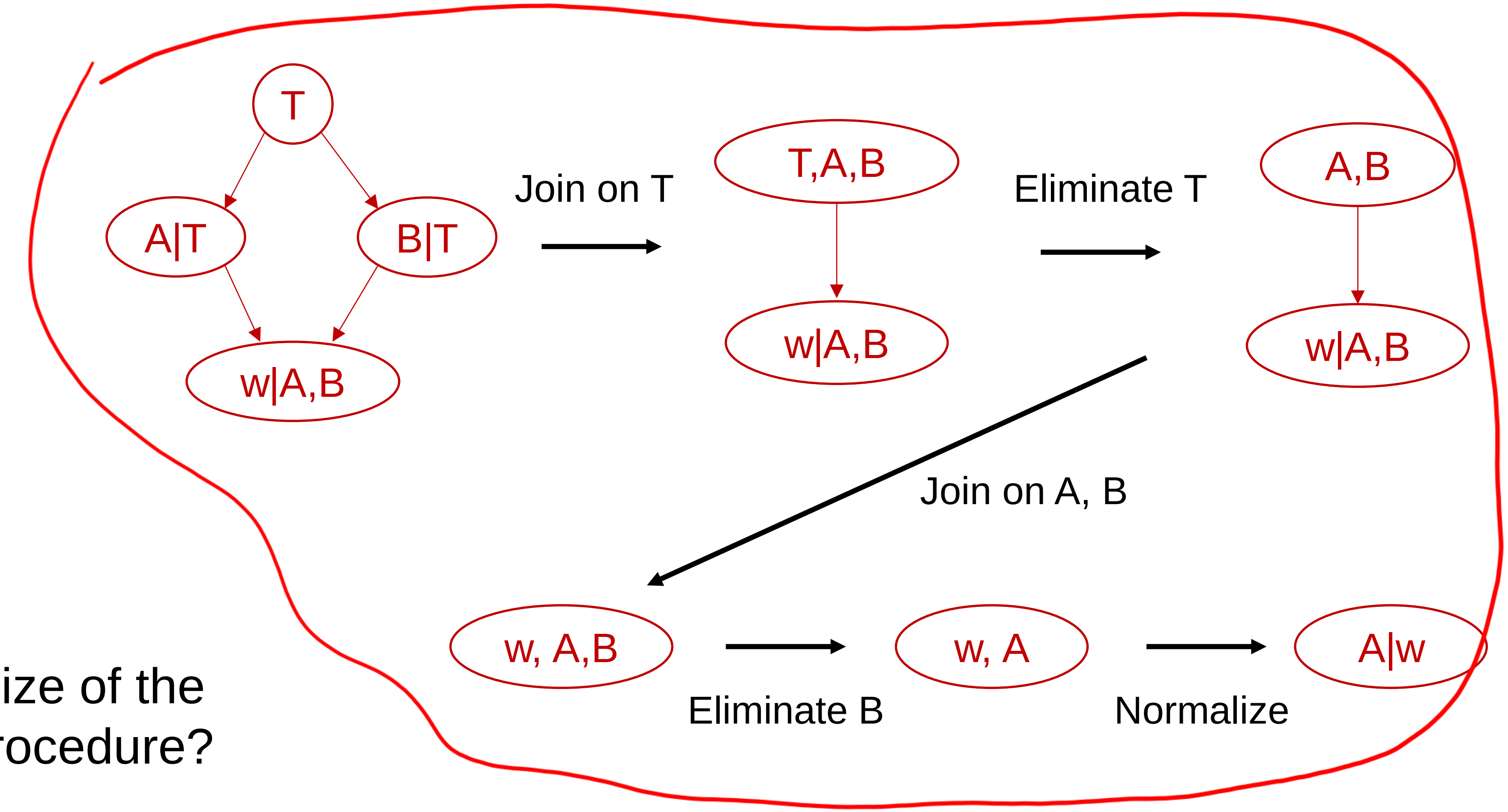
Learning Objective 2

- **Inference:** Given a Bayesian network, a query variable Q, and evidences $\{E_1 = e_1, \ldots, E_k = e_k\}$, calculate $P(Q|E_1 = e_1, \ldots, E_k = e_k)$
- Exact Inference in Bayesian networks
- Variable elimination
  - Only keep entries consistent with the evidence variable
  - Interleave **Join** and **Marginalize (Eliminate)** operations

Learning Objective 2 – Example

In the previous example, how to calculate the conditional probability P(A=bad | w=A)?





What's the maximum size of the table involved in this procedure?

Learning Objective 2 – Example

T	P(T)
Warm	0.4
Cold	0.6

T	A	P(A|T)
Warm	Good	0.6
Warm	Bad	0.4
Cold	Good	0.5
Cold	Bad	0.5

T	B	P(B|T)
Warm	Good	0.8
Warm	Bad	0.2
Cold	Good	0.3
Cold	Bad	0.7

Join on T →

P(T) P(A|T) P(B|T)

=

T	A	B	P(T,A,B)
Warm	Good	Good	0.4 * 0.6 * 0.8
Warm	Good	Bad	0.4 * 0.6 * 0.2
Warm	Bad	Good	0.4 * 0.4 * 0.8
Warm	Bad	Bad	0.4 * 0.4 * 0.2
Cold	Good	Good	0.6 * 0.5 * 0.3
Cold	Good	Bad	0.6 * 0.5 * 0.7
Cold	Bad	Good	0.6 * 0.5 * 0.3
Cold	Bad	Bad	0.6 * 0.5 * 0.7

Learning Objective 2 – Example

T	A	B	P(T,A,B)
Warm	Good	Good	0.4 * 0.6 * 0.8
Warm	Good	Bad	0.4 * 0.6 * 0.2
Warm	Bad	Good	0.4 * 0.4 * 0.8
Warm	Bad	Bad	0.4 * 0.4 * 0.2
Cold	Good	Good	0.6 * 0.5 * 0.3
Cold	Good	Bad	0.6 * 0.5 * 0.7
Cold	Bad	Good	0.6 * 0.5 * 0.3
Cold	Bad	Bad	0.6 * 0.5 * 0.7

Eliminate (Marginalize) T →

A	B	P(A,B)
Good	Good	0.282
Good	Bad	0.258
Bad	Good	0.218
Bad	Bad	0.242

Learning Objective 2 – Example

A	B	P(A,B)
Good	Good	0.282
Good	Bad	0.258
Bad	Good	0.218
Bad	Bad	0.242

Join on A,B

→

A	B	W	P(W|A,B)
Good	Good	A	0.4
~~Good~~	~~Good~~	~~B~~	~~0.6~~
Good	Bad	A	0.8
~~Good~~	~~Bad~~	~~B~~	~~0.2~~
Bad	Good	A	0.2
~~Bad~~	~~Good~~	~~B~~	~~0.8~~
Bad	Bad	A	0.4
~~Bad~~	~~Bad~~	~~B~~	~~0.6~~

Learning Objective 2 – Example

A	B	P(A,B)
Good	Good	0.282
Good	Bad	0.258
Bad	Good	0.218
Bad	Bad	0.242

A	B	W	P(W|A,B)
Good	Good	A	0.4
Good	Bad	A	0.8
Bad	Good	A	0.2
Bad	Bad	A	0.4

Join on A,B →

$P(A,B)\,P(W|A,B)$

=

A	B	W	P(A,B,W)
Good	Good	A	0.1128
Good	Bad	A	0.2064
Bad	Good	A	0.0436
Bad	Bad	A	0.0968

Learning Objective 2 – Example

A	B	W	P(A,B,W)
Good	Good	A	0.1128
Good	Bad	A	0.2064
Bad	Good	A	0.0436
Bad	Bad	A	0.0968

Eliminate B →

A	W	P(A,W)
Good	A	0.1128 + 0.2064
Bad	A	0.0436 + 0.0968

Learning Objective 2 – Example

A	W	P(A,W)
Good	A	0.3192
Bad	A	0.1404

Normalize →

A	P(A | w=A)
Good	0.6945
Bad	0.3055

Learning Objective 3

- Approximate inference in Bayesian network
  - Prior sampling
  - Rejection sampling
  - Likelihood weighting
  - Gibbs sampling

Learning Objective 3 – Example

- In the previous example, suppose that we know A wins, and we have the following samples drawn from the Bayesian network:
  - (T=Warm, A=Bad, B=Good), (T=Warm, A=Good, B=Good), (T=Cold, A=Bad, B=Bad)
- Use Likelihood weighting to estimate P(A=bad | w=A)

T	A	B	Weight = P(w=A | A,B)
Warm	Bad	Good	0.2
Warm	Good	Good	0.4
Cold	Bad	Bad	0.4

Normalize →

$$P_{est}(A=Bad \mid w=A) = \frac{0.2+0.4}{0.2+0.4+0.4} = 0.6$$

Recall:
If both team are in the same condition, B wins with probability 0.6
Otherwise, the good-conditioned team wins with probability 0.8

Gibbs sampling algorithm

- Repeat many times: Sample a non-evidence variable X_i from

  $P(X_i \mid x_1,..,x_{i-1},x_{i+1},..,x_n)$

  $= P(X_i \mid \text{Markov\_blanket}(X_i))$

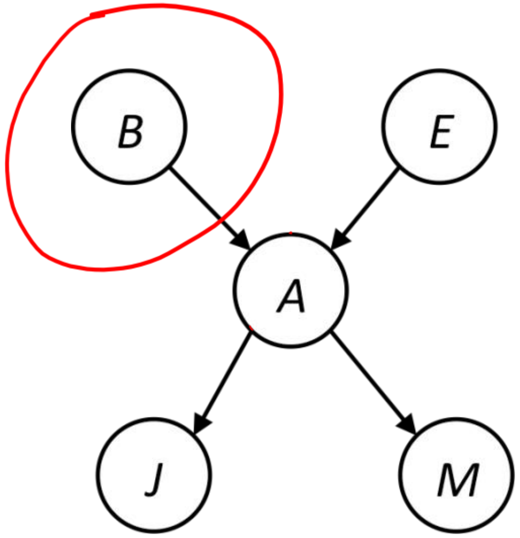
  $= \alpha\ P(X_i \mid \text{Parents}(X_i)) \prod_j P(y_j \mid \text{Parents}(Y_j))$

  children(X_i) involve X_i

Page 74 in BN slides has an example illustrating why we only need to consider Markov blanket of X_i

- Markov_blanket(X_i) includes
  - X_i's parents
  - X_i's children A
  - X_i's children's parent E

$P(B \mid E, A, J, M) = P(B \mid E, A)$



Markov_blanket(B) = ?

(Hidden) Markov Model

Learning Objective 1

- Exact inference for Markov models
  - Forward algorithm (= repeatedly Join and Eliminate)

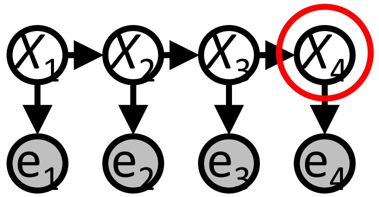
$$\begin{cases} P(X_1) \\ P(X_{i+1} \mid X_i) \end{cases}$$

$$\boxed{P(X_i)}$$

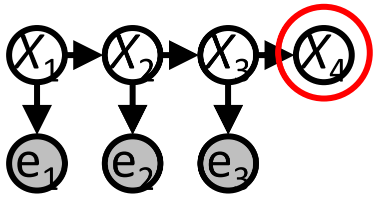
Learning Objective 2

- Exact inference for Hidden Markov models
  - Filtering
  - Prediction
  - Smoothing (forward-backward algorithm)
  - Most-likely Sequence (Viterbi algorithm)

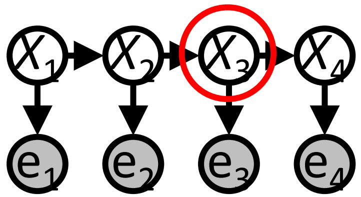
Filtering: $P(X_t|e_{1:t})$



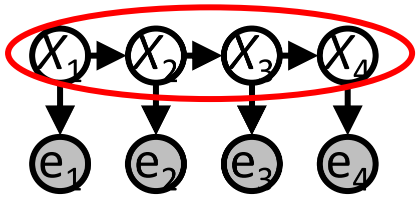
Prediction: $P(X_{t+k}|e_{1:t})$



Smoothing: $P(X_k|e_{1:t})$, $k<t$

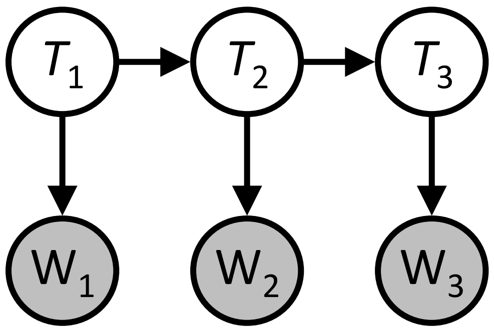


Most likely seq: argmax $P(X_{1:t}|e_{1:t})$



Learning Objective 2 – Example

- Smoothing



T_1	$P(T_1)$
Warm	0.5
Cold	0.5

T_i	T_{i+1}	$P(T_{i+1}\|T_i)$
Warm	Warm	0.8
Warm	Cold	0.2
Cold	Warm	0.3
Cold	Cold	0.7

T_i	W_i	$P(W_i\|T_i)$
Warm	A	0.4
Warm	B	0.6
Cold	A	0.7
Cold	B	0.3

$$P(T_2 \mid w_1 = A, w_2 = A, w_3 = A) = ?$$

$$P(T_2 \mid w_1 = A, w_2 = A, w_3 = A) \propto P(T_2, w_3 = A \mid w_1 = A, w_2 = A)$$

$$= \underbrace{P(T_2 \mid w_1 = A, w_2 = A)}_{\text{filtering}} \underbrace{P(w_3 = A \mid T_2)}$$

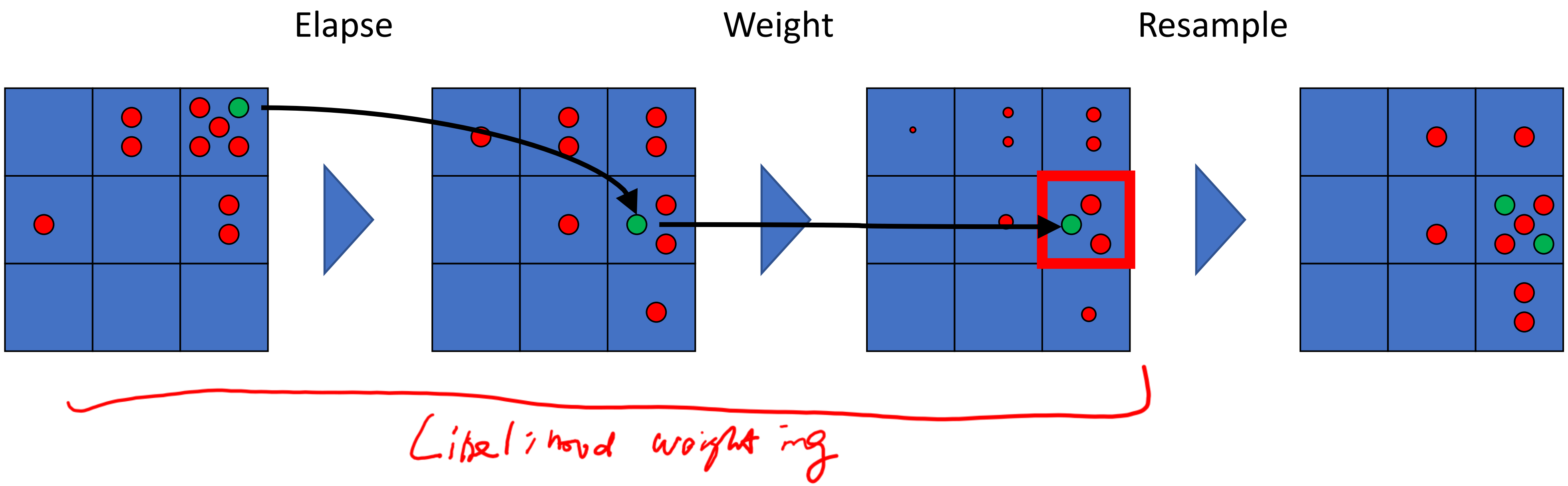
filtering

$$= \sum_t P(T_3 = t \mid T_2) P(w_3 = A \mid T_3 = t)$$

See Page 33-34 in HMM slides for the more general case (i.e., longer sequence)

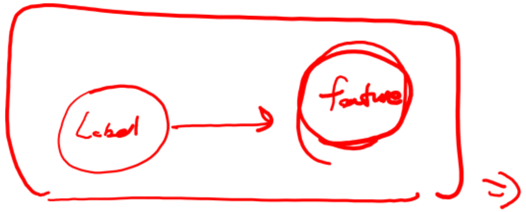
Learning Objective 3

- Approximate inference for Hidden Markov models
  - Particle filtering (likelihood weighting + resampling)

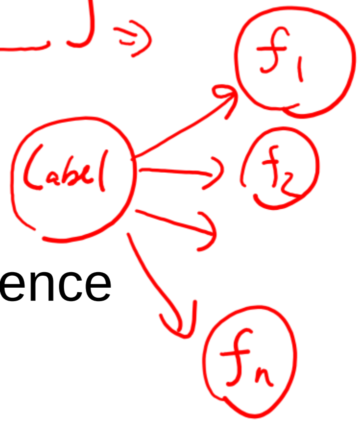


Machine Learning

Learning Objective 1



- Naïve Bayes
  - The modeling assumption: features are mutually independent
  - Maximum likelihood estimation – simply counting the feature occurrence
  - Regularization: Laplace smoothing



Learning Objective 1 – Example

- Suppose that we use Naïve Bayes to conduct **spam filtering**. We use **Laplace smoothing** with k=1 for regularization (i.e., give every word a fake count of 1).
- Training data:
  - **Spam**, “Winner!! As a valued customer you have been selected to receive a $900 prize reward!!!”
  - **Spam**, “We are trying to contact you. Last weekend’s draw shows that you won a £1000 prize guaranteed!!!”
  - **Ham**, “Hey! Did you want to grab coffee before the team meeting on Friday?”
  - **Ham**, ““Thank you for attending the talk this morning. I’ve attached the presentation for you to share with your team.”
- In the learned model, what are P(Spam), P(“prize” | Spam), P(“prize” | Ham)?

See the corrected answer in the next slide

Learning Objective 1 – Example

- Suppose that we use Naïve Bayes to conduct **spam filtering**. We use **Laplace smoothing** with k=1 for regularization (i.e., give every word a fake count of 1).
- Training data:
  - **Spam**, "Winner!! As a valued customer you have been selected to receive a $900 prize reward!!!"
  - **Spam**, "We are trying to contact you. Last weekend's draw shows that you won a £1000 prize guaranteed!!!"
  - **Ham**, "Hey! Did you want to grab coffee before the team meeting on Friday?"
  - **Ham**, ""Thank you for attending the talk this morning. I've attached the presentation for you to share with your team."
- In the learned model, what are P(Spam), P("prize" | Spam), P("prize" | Ham)?

P(Spam) = $\frac{2}{4}$ P("price"|Spam) = $\frac{2+k\cdot 1}{2+k\cdot 2} = \frac{3}{4}$ P("price"|Ham) = $\frac{0+k\cdot 1}{2+k\cdot 2} = \frac{1}{4}$

Learning Objective 1 – Example

- Now we get a new email and want to classify it into spam or ham.
- Suppose for simplicity it only contains "hey customer".
- How should we classify it?

$T \in \{spam, ham\}$

P(T | "hey customer") ∝ P(T) × P("hey customer" | T)

= P(T) × P("hey" | T) × P("customer" | T)

Let α = P(Spam) × P("hey" | Spam) × P("customer" | Spam)

β = P(Ham) × P("hey" | Ham) × P("customer" | Ham)

Then P(Spam | "hey customer") = $\frac{\alpha}{\alpha+\beta}$

Learning Objective 2

- Logistic regression
  - The modeling assumption
  - Stochastic gradient descent
  - Regularization

Learning Objective 2 – Example

- Suppose that we use logistic regression for a classification problem
- Feature dimension = 2 and #Classes = 3
  - What's the number of parameters in the model? $3 \times 2 = 6$
- Suppose that the final model we get after training is
  - $w^{(1)}$ = [0.7, -0.1]
  - $w^{(2)}$ = [0.3, -0.4] ✓
  - $w^{(3)}$ = [-0.9, 0.6]
  - Then for an input feature x = [0,1], what's the $P(Y \mid x)$ given by this model?

$$\frac{e^{-0.1}}{e^{-0.1} + e^{-0.4} + e^{0.6}}$$

$$score(class\ 1, x) = w^{(1)} \cdot x = [0.7, -0.1] \cdot [0, 1] = -0.1$$

$$score(class\ 2, x) = -0.4$$

$$score(class\ 3, x) = 0.6$$

$$P(class\ 1 \mid x) = \frac{\exp(score(class\ 1, x))}{\sum_{i=1}^{3} \exp(score(class\ i, x))}$$

Learning Objective 3

- General considerations in Machine Learning
  - Overfitting and regularization
  - Hyperparameter: quantities chosen in the training procedure
  - Hyperparameter tuning
- Hyperparameter tuning
  - Split the original dataset into training / held-out / test datasets
  - Run machine learning algorithm (e.g., SGD) on the training dataset with multiple hyperparameters, which gives multiple models
  - Choose the model that has the best performance on the held-out dataset
  - Report the performance of the model on the test dataset

chosen

Deep Learning and Applications

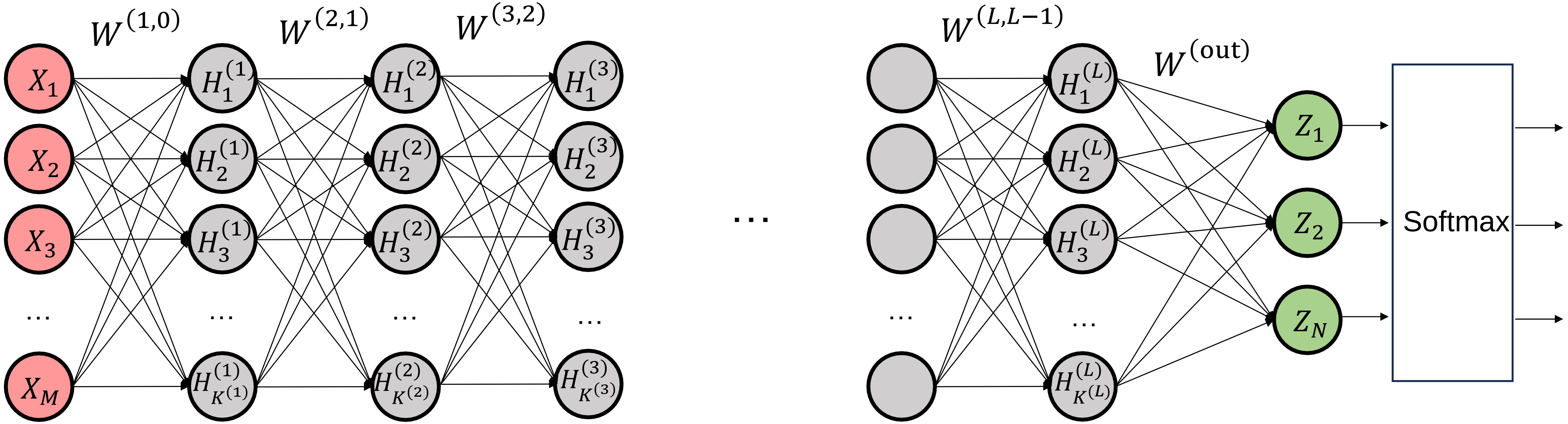
Learning Objective 1

- Components of neural networks
  - Hidden layers
  - Activation functions
  - Softmax layer (for classification problems)

$$H_i^{(0)} \coloneqq X_i$$

$$H_i^{(\ell)} = g\left(\sum_j W_{ij}^{(\ell,\ell-1)} H_j^{(\ell-1)}\right) \quad \forall \ell = 1, \dots, L$$

$$Z_i = \sum_j W_{ij}^{(\text{out})} H_j^{(L)}$$



Learning Objective 2

- High-level understandings about the techniques mentioned in the guest lectures
- Computer vision applications
  - Object detection
  - R-CNN, Fast R-CNN, and Faster R-CNN
- Natural language processing applications
  - Word vector
  - Recurrent neural network
  - Transformer

Reinforcement Learning

Learning Objective

- Value iteration under known MDP model
- Q-Learning under unknown MDP model

Value Iteration

$V_0(s) \leftarrow 0 \quad \forall s$

For $k = 1, 2, \ldots$

$$Q_k(s,a) \leftarrow \sum_{s'} T(s,a,s')[R(s,a,s') + \underbrace{\gamma V_{k-1}(s')}_{=0 \text{ if } s \text{ is terminal state}}] \quad \forall s,a$$

$$V_k(s) \leftarrow \max_a Q_k(s,a) \quad \forall s$$

If $|V_k(s) - V_{k-1}(s)| \leq \epsilon$ for all s:

Let $\hat{Q}(s,a) = Q_k(s,a) \ \forall s,a$

break

Return policy $\hat{\pi}(s) = \underset{a}{\operatorname{argmax}}\, \hat{Q}(s,a)$

Value Iteration – Example

s	a	s'	T(s,a,s’)	R(s,a,s’)
	Slow		1.0	+1
	Fast		0.5	+2
	Fast		0.5	+2
	Slow		0.5	+1
	Slow		0.5	+1
	Fast		1.0	–10
	(end)		1.0	0

			
V_2	3.5	2.5	0
V_1	2	1	0
V_0	0	0	0

Assume no discount $(\gamma = 1)$

Q-Learning

$V_0(s) \leftarrow 0,\ Q_0(s,a) \leftarrow 0 \quad \forall s,a$

Let s_1 be the initial state.

For $k = 1, 2, \ldots$ Epsilon-Greedy strategy or Boltzmann exploration strategy

Take action a_k. Observe next state s_{k+1} and reward $R_k = R(s_k, a_k, s_{k+1})$.

// Slightly modify the values on the visited state-action pair (s_k, a_k):

$Q_k(s_k, a_k) \leftarrow (1 - \eta_k)\, Q_{k-1}(s_k, a_k) + \eta_k \left[R_k + \gamma V_{k-1}(s_{k+1})\right]$ $\eta_k \in (0,1)$: learning rate

$V_k(s_k) \leftarrow \max_a Q_k(s_k, a)$ =0 if s_k is terminal

// Keep other values unchanged:

$Q_k(s,a) \leftarrow Q_{k-1}(s,a)$ and $V_k(s) \leftarrow V_{k-1}(s)$ for $(s,a) \neq (s_k, a_k)$

If s_k is a terminal state:

Reset s_{k+1} to be the initial state.

Continue

Q-Learning Example

- Page 59-61 in the RL lecture slides

Course Evaluation

- Ending on December 9 (4 days from now!)
- To encourage you to respond, we give 1.5 extra points to students finishing it
- The way to calculate the final score:
  - Calculate the raw scores
  - Set score thresholds (e.g., adjusting the percentage of students getting A, B, etc.)
  - Add the 1.5 extra points

Thank you

- Good luck for your finals!