# Lecture 16: Exam 1 Review

Mark Hasegawa-Johnson

# Outline

- How to take the exam
- What's on the exam
- Sample problems

# How to take the exam

- Go to https://cbtf.Illinois.edu, choose CS440/ECE448 exam 1, and register the time and location where you will take the exam
- If you can’t register at cbtf.Illinois.edu, contact us on campuswire

# How to take the exam

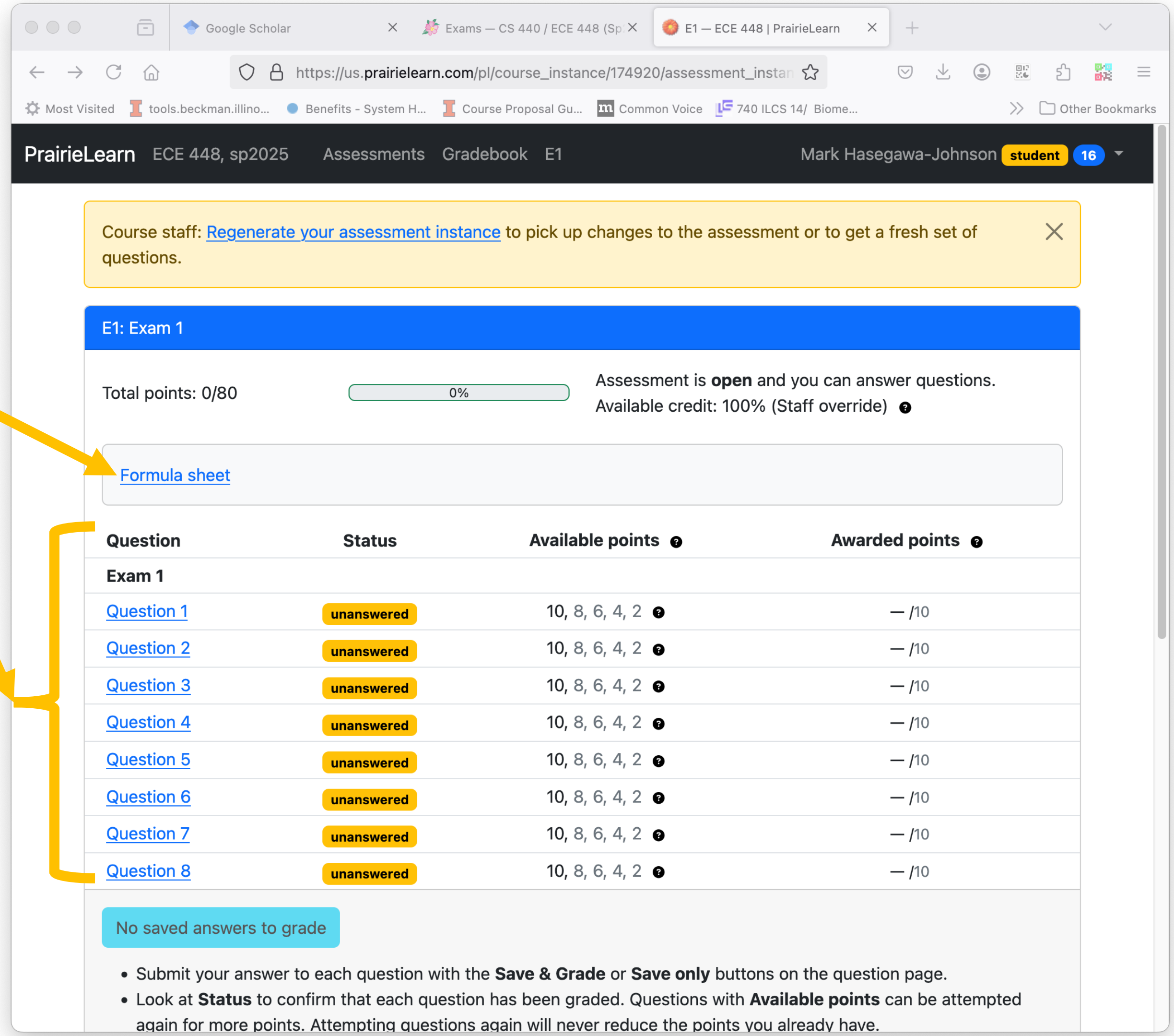
- Show up on time at the appointed location! It's possible to reschedule if you miss an exam, but it will require you to go to a CBTF location to apply in person for permission to do so.
- Bring: Pencils and erasers
- CBTF will provide: Scratch paper
- The exam will have attached: a PDF formula sheet, which is also available now on the course web page, so you can see what will be on it

# Exam format

- The exam will have 8 questions.
- Most or all will be questions that you have seen in daily quizzes. Most are multiple choice, though some are short-answer.
- Points per question: 10,8,6,4,2 if you get it right on the first, second, third, fourth, fifth try.

# Exam format

- Formula sheet
- Questions to answer



# Outline

- How to take the exam
- What's on the exam
- Sample problems

# Topics

- Probability
- Decision theory and fairness
- Naïve bayes
- Learning and linear regression
- Logistic regression and word2vec
- Nonlinear regression and relevance backprop
- Softmax and transformer
- HMMs and Bayesian networks

# Probability

- Axioms of probability:

$$P(A) \geq 0, P(\Omega) = 1, P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

- Jointly random variables

$$P(X = \boldsymbol{x}) = P(X_1 = x_1, \cdots, X_n = x_n)$$

- Conditional Probability

$$P(X, Y) = P(X|Y)P(Y)$$

- Independence

$$P(X|Y) = P(X) \Leftrightarrow P(X, Y) = P(X)P(Y)$$

- Expectation

$$E[f(X, Y)] = \sum_{x,y} f(x, y)P(X = x, Y = y)$$

# Decision theory

- Minimum Probability of Error = Maximum *A Posteriori* Probability*:*
$$f(x) = \underset{y}{\operatorname{argmax}} P(Y = y|X = x)$$
- Bayes Error Rate:
$$\text{Bayes Error Rate} = \sum_x P(X = x) \min_y P(Y \neq y|X = x)$$
- Confusion Matrix, Precision & Recall, Sensitivity & Selectivity
$$\text{Precision} = P(Y = 1|f(X) = 1) = \frac{TP}{TP + FP}$$
$$\text{Sensitivity} = \text{Recall} = P(f(X) = 1|Y = 1) = \frac{TP}{TP + FN}$$
$$\text{Selectivity} = P(f(X) = 0|Y = 0) = \frac{TN}{TN + FP}$$
- Train, Dev, and Test Corpora

# Naïve Bayes

- MPE = MAP with Bayes' rule:

$$f(x) = \underset{y}{\operatorname{argmax}}(\log P(Y = y) + \log P(X = x|Y = y))$$

- naïve Bayes:

$$\log P(X = x|Y = y) \approx \sum_{i=1}^{n} \log P(W = w_i|Y = y)$$

- maximum likelihood parameter estimation:

$$P(W = w_i|Y = y) = \frac{\text{Count}(w_i, y)}{\sum_{v \in V} \text{Count}(v, y)}$$

- Laplace Smoothing:

$$P(W = w_i|Y = y) = \begin{cases} \dfrac{k + \text{Count}(w_i, y)}{k + \sum_{v \in V}(k + \text{Count}(v, y))} & W = OOV \text{ is possible} \\ \dfrac{k + \text{Count}(w_i, y)}{\sum_{v \in V}(k + \text{Count}(v, y))} & \text{otherwise} \end{cases}$$

# Three Definitions of Fairness

- Demographic Parity: $P(f(X)|A=1) = P(f(X)|A=0)$
- Equal Odds: $P(f(X)|Y, A=1) = P(f(X)|Y, A=0)$
- Predictive Parity: $P(Y|f(X), A=1) = P(Y|f(X), A=0)$

Those three things can only all be true, all at the same time, if:

- $P(Y|A=1) = P(Y|A=0)$

# Learning

- **Biological inspiration:** Neurons that fire together wire together. Given enough training examples $(x_i, y_i)$, can we learn a desired function so that $f(x) \approx y$?
- **Classification tree:** Learn a sequence of if-then statements that computes $f(x) \approx y$
- **Mathematical definition of supervised learning:** Given a training dataset, $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$, find a function $f$ that minimizes the risk, $\mathcal{R} = \mathrm{E}[\ell(Y, f(X))]$.
- **Overtraining:** $\mathcal{R}_{\text{emp}} = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i))$ reaches zero if you train long enough.
- **Early Stopping**: Stop when error rate on the dev set reaches a minimum

# Linear Regression

- Definition of linear regression

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b$$

- Mean-squared error

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_i\,, \qquad \mathcal{L}_i = \frac{1}{2}\epsilon_i^2, \qquad \epsilon_i = f(\boldsymbol{x}_i) - y_i$$

- Gradient descent

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}}, \qquad \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = \frac{1}{n}\sum_{i=1}^{n} \epsilon_i \boldsymbol{x}_i$$

- Stochastic gradient descent

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta\frac{\partial \mathcal{L}_i}{\partial \boldsymbol{w}}, \qquad \frac{\partial \mathcal{L}_i}{\partial \boldsymbol{w}} = \epsilon_i \boldsymbol{x}_i$$

# Logistic Regression

- Logistic regression

$$f(\boldsymbol{x}_i) = \sigma(\boldsymbol{x}_i^T \boldsymbol{w}) = \frac{1}{1 + e^{-\boldsymbol{x}_i^T \boldsymbol{w}}}$$

- Derivative of the sigmoid

$$\frac{\partial \sigma(z)}{\partial z} = \sigma(z)(1 - \sigma(z))$$

- Derivative of the negative log sigmoid

$$\mathcal{L} = -\sum_{y_i=1} \log \sigma\left(\boldsymbol{x}_i^T \boldsymbol{w}\right) - \sum_{y_i=0} \log\left(1 - \sigma(\boldsymbol{x}_i^T \boldsymbol{w})\right)$$

# word2vec

- Continuous bag of words:

$$\mathcal{L} = -\frac{1}{T}\sum_{t=0}^{T-1}\sum_{j=-c,j\neq 0}^{c} \ln P\left(w_t | w_{t+j}\right)$$

- Skip-gram:

$$\mathcal{L} = -\frac{1}{T}\sum_{t=0}^{T-1}\sum_{j=-c,j\neq 0}^{c} \ln P\left(w_{t+j} | w_t\right)$$

- Dot-product similarity:

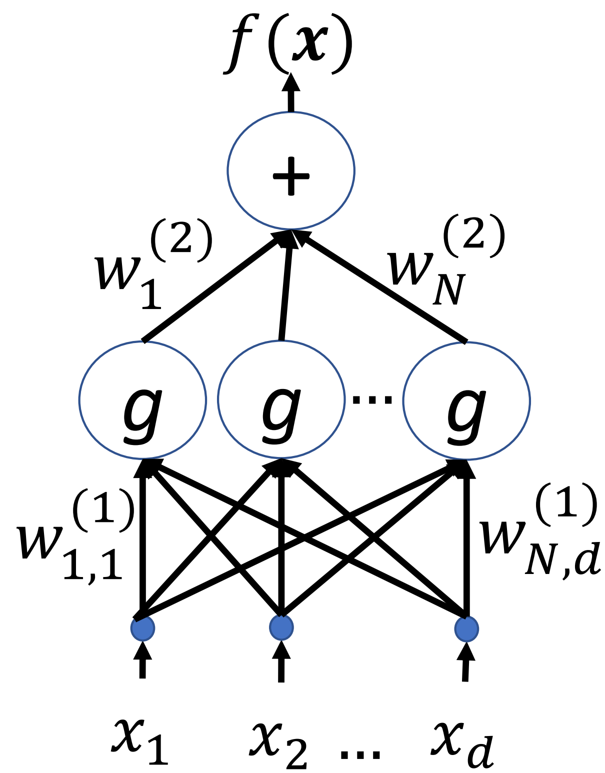
$$\mathcal{L} = -\frac{1}{T}\sum_{t=0}^{T-1}\sum_{j=-c,j\neq 0}^{c} \ln \sigma(\boldsymbol{v}_1^T \boldsymbol{v}_2)$$

- Skip-gram noise contrastive estimation:

$$\mathcal{L} = -\ln \sigma\left(\boldsymbol{v}_{t+j}^T \boldsymbol{v}_t\right) - \frac{1}{k}\sum_{i=1}^{k} \ln\left(1 - \sigma(\boldsymbol{v}_i^T \boldsymbol{v}_t)\right)$$

# Nonlinear regression

- Piece-wise constant nonlinear regression:
$$f(\boldsymbol{x}) = \boldsymbol{w}^{(2),T}\sigma\left(\boldsymbol{W}^{(1)}\boldsymbol{x}\right)$$
- Piece-wise linear regression:
$$f(\boldsymbol{x}) = \boldsymbol{w}^{(2),T}\mathrm{ReLU}\left(\boldsymbol{W}^{(1)}\boldsymbol{x}\right)$$
- Back-propagation:
$$\boldsymbol{W}^{(1)} \leftarrow \boldsymbol{W}^{(1)} - \eta\frac{\partial \mathcal{L}_i}{\partial \boldsymbol{W}^{(1)}}$$
$$\boldsymbol{w}^{(2)} \leftarrow \boldsymbol{w}^{(2)} - \eta\frac{\partial \mathcal{L}_i}{\partial \boldsymbol{w}^{(2)}}$$



# Softmax

- Multi-class linear classifiers

$$f(\boldsymbol{x}) = \operatorname{argmax} \boldsymbol{W}\boldsymbol{x}$$

- One-hot vectors

$$f_c(\boldsymbol{x}) = \mathbb{1}_{\operatorname{argmax} \boldsymbol{W}\boldsymbol{x} = c}$$

- Softmax nonlinearity

$$f_c(\boldsymbol{x}) = \frac{\exp(\boldsymbol{w}_c^T \boldsymbol{x})}{\sum_{k=1}^{v} \exp(\boldsymbol{w}_k^T \boldsymbol{x})}$$

- Derivative of the log softmax

$$\boldsymbol{w}_c \leftarrow \boldsymbol{w}_c + \eta \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{n}} \left( \mathbb{1}_{y_i = c} - f_c(\boldsymbol{x}) \right) \boldsymbol{x}_i$$

# Transformer

- Query, Key, and Value matrices

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_1^T \\ \vdots \\ \boldsymbol{v}_n^T \end{bmatrix}, \boldsymbol{K} = \begin{bmatrix} \boldsymbol{k}_1^T \\ \vdots \\ \boldsymbol{k}_n^T \end{bmatrix}, \boldsymbol{Q} = \begin{bmatrix} \boldsymbol{q}_1^T \\ \vdots \\ \boldsymbol{q}_m^T \end{bmatrix}$$

- Attention

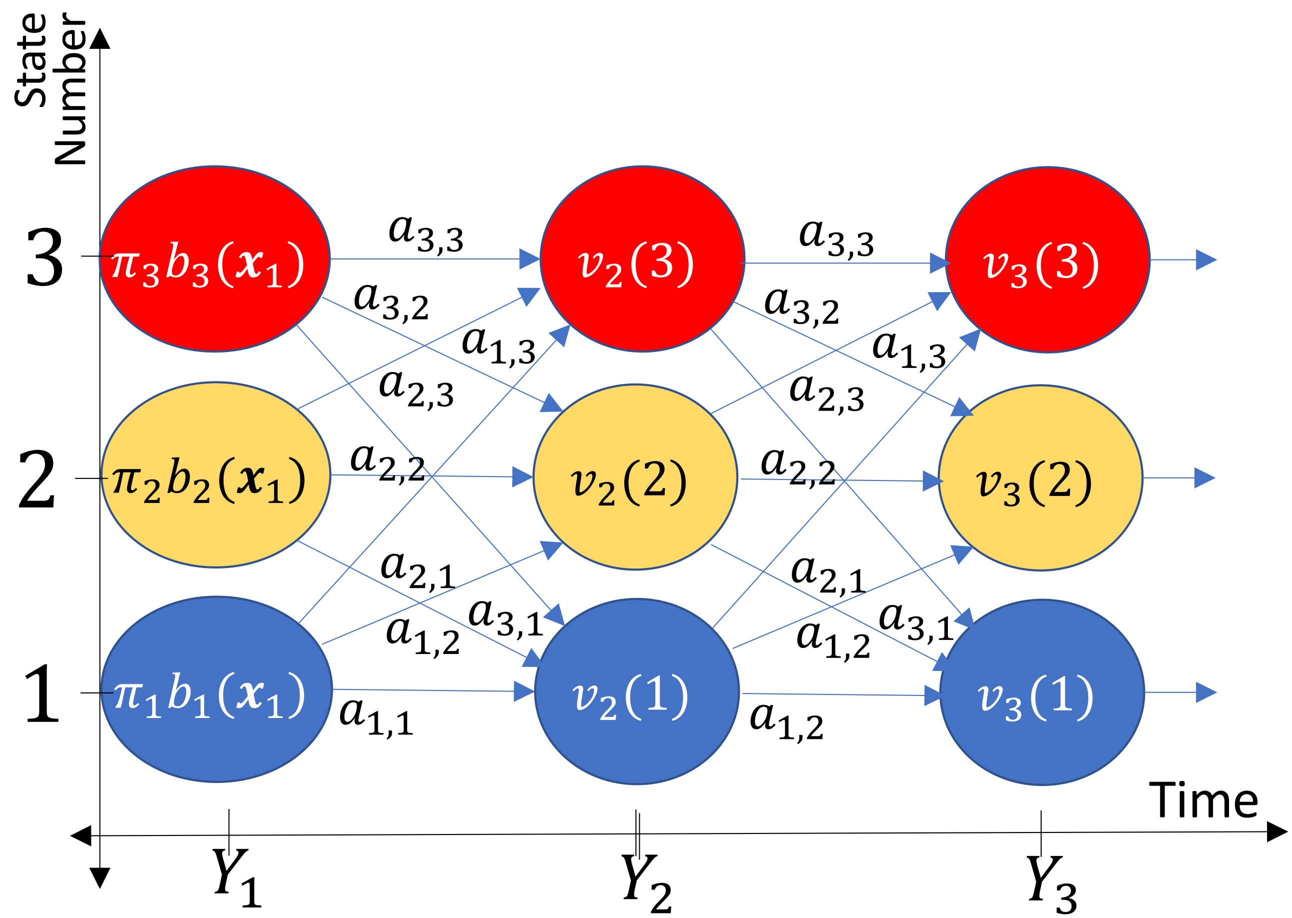
$$\boldsymbol{c}_i = \boldsymbol{V}^{\boldsymbol{T}} \mathrm{softmax}(\boldsymbol{K}\boldsymbol{q}_i) = \sum_t \frac{\exp\left(\boldsymbol{q}_i^T \boldsymbol{k}_t\right)}{\sum_\tau \exp\left(\boldsymbol{q}_i^T \boldsymbol{k}_\tau\right)} \boldsymbol{v}_t$$

- Positional encoding

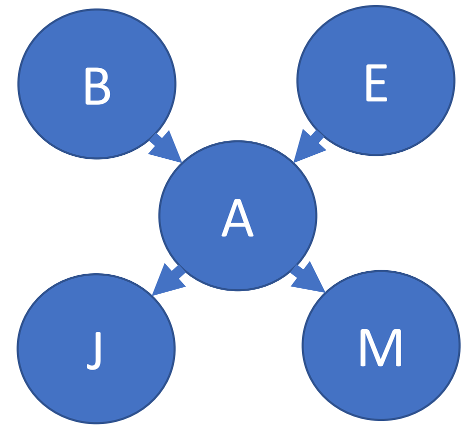
$$\boldsymbol{x}_t \mathrel{+}= \begin{bmatrix} \cos\left(\frac{\pi t}{T}\right) \\ \sin\left(\frac{\pi t}{T}\right) \\ \vdots \end{bmatrix}$$

# HMM: Viterbi Algorithm

$$v_t(j) = \max_i v_{t-1}(i)\, a_{i,j} b_j(\boldsymbol{x}_t)$$
$$\psi_t(j) = \operatorname{argmax}_i v_{t-1}(i)\, a_{i,j} b_j(\boldsymbol{x}_t)$$



# Bayesian Networks




- Bayesian network: A better way to represent knowledge
  - Reduces space complexity from $\mathcal{O}\{v^n\}$ to $\mathcal{O}\{nv^p\}$
  - Does not automatically reduce time complexity.
- Key ideas: Independence and Conditional independence

| | | Shared Ancestor (of at least one)? | |
|---|---|---|---|
| | | **No** | **Yes** |
| **Shared Descendant (of both)?** | **No** | Independent | Dependent unless shared ancestor value is known |
| | **Yes** | Independent unless shared descendant value is known | Dependent unless shared ancestor value known AND shared descendant value unknown |

# Explainable AI

- Relevance of input $x_d$ to output $z_c$:

$$R_{c,d} = \frac{\partial z_c}{\partial x_d} \cdot x_d$$

  - Layer-wise relevance propagation: Normalize so $\sum_d R_{c,d} = R_c$
  - Gradient-weighted class activation mapping: Keep only positive relevances
- Counter-Factual Reasoning:

$$\max_{F,X} |P(F, X, A = a) - P(F, X, A = \neg a)| > \epsilon?$$

  - If the answer is yes, then you should add an edge between A and F

# Outline

- How to take the exam
- What’s on the exam
- Sample problems

# Sample problems

- You can do the daily quizzes again, as often as you like
  - Only your highest score counts toward your grade (presumably this is the one you received the first time you did the quiz, when 100% was possible)
  - After you get a problem correct, you can click “Try a new variant” to try a new variant
- Sample exam is available
  - Coverage is similar to the exam next week
  - Style of questions is quite different

# Sample problems

- Sample problems
  - NOT the same format as the real exam
- Formula sheet
  - SAME as the real exam