

# **CS195-5 : Introduction to Machine Learning**

## **Lecture 35**

Greg Shakhnarovich

December 8, 2006

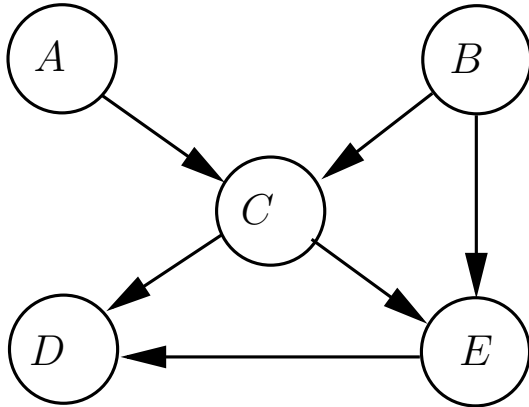
---

# Announcements

- Next lectures in the usual location (Lubrano).
  - Monday 12/11: advanced applications.
  - Wednesday 12/13: final review.
  - Final: Monday 12/18, Wilson 101, 9am-noon.
  - 200 level projects due December 31st.

---

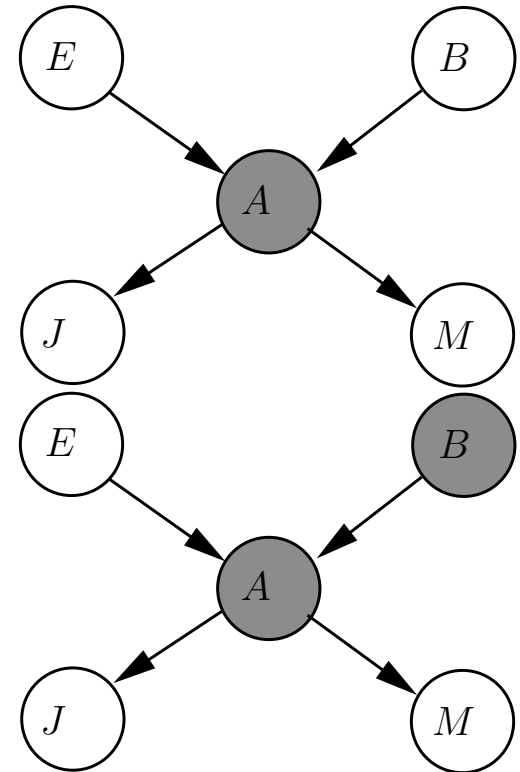
## Review: directed graphical models



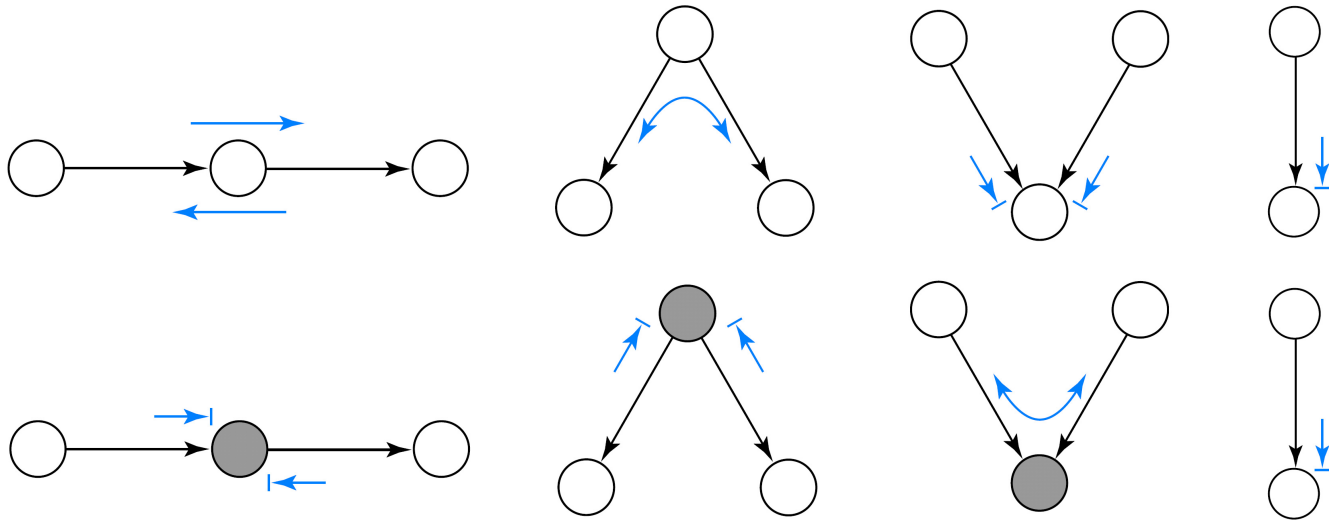
$$p(A)p(B)p(C | A, B)p(E | B, C)p(D | E, C)$$

# Review: inference

- Induced dependence/independence
- Explaining away



# Review: the Bayes Ball algorithm



---

# What have we covered?

- Supervised learning: regression and classification.
- Unsupervised learning: clustering, dimensionality reduction, estimation.
- Probabilistic models: Naive Bayes, mixture models, Markov and HMM,
- Graphical models
- Theory: bias/variance, max-margin classifiers.

---

# Evaluation in machine learning

- How do you evaluate a classification or regression method?
  - How do we compare two algorithms/models?
  - What do we report?
- Idea 1: training set.

---

# Evaluation in machine learning

- How do you evaluate a classification or regression method?
  - How do we compare two algorithms/models?
  - What do we report?
- Idea 1: training set.
- Idea 2 (better): cross-validation.

---

# Evaluation in machine learning

- How do you evaluate a classification or regression method?
  - How do we compare two algorithms/models?
  - What do we report?
- Idea 1: training set.
- Idea 2 (better): cross-validation. Problems:
  - ignores dependencies across folds,
  - overutilizes the data.

---

# Evaluation in machine learning

- How do you evaluate a classification or regression method?
  - How do we compare two algorithms/models?
  - What do we report?
- Idea 1: training set.
- Idea 2 (better): cross-validation. Problems:
  - ignores dependencies across folds,
  - overutilizes the data.
- Idea 3 (if can afford it): test set.

---

## Evaluation on a test set

- Suppose you trained a classifier  $A \Rightarrow$  on 10 examples,  $\hat{\epsilon}_A = 0.1$
- Another classifier  $B \Rightarrow$  test error  $\hat{\epsilon}_B = 0.2$
- Which one is better?

---

## Evaluation on a test set

- Suppose you trained a classifier  $A \Rightarrow$  on 10 examples,  $\hat{\epsilon}_A = 0.1$
- Another classifier  $B \Rightarrow$  test error  $\hat{\epsilon}_B = 0.2$
- Which one is better? Naive approach:  $A$  since  $\hat{\epsilon}_A < \hat{\epsilon}_B$ .
- What if in reality, the expected risk is  $\epsilon_A = 0.3$  and  $\epsilon_B = 0.015$ ?
  - Probability of observing  $\hat{\epsilon}_A$  is

$$\text{Binomial}_{0.3}(1, 10; ) = 0.1211,$$

---

## Evaluation on a test set

- Suppose you trained a classifier  $A \Rightarrow$  on 10 examples,  $\hat{\epsilon}_A = 0.1$
- Another classifier  $B \Rightarrow$  test error  $\hat{\epsilon}_B = 0.2$
- Which one is better? Naive approach:  $A$  since  $\hat{\epsilon}_A < \hat{\epsilon}_B$ .
- What if in reality, the expected risk is  $\epsilon_A = 0.3$  and  $\epsilon_B = 0.015$ ?
  - Probability of observing  $\hat{\epsilon}_A$  is

$$\text{Binomial}_{0.3}(1, 10;) = 0.1211,$$

and probability of observing  $\hat{\epsilon}_B$  is

$$\text{Binomial}_{0.1}(2, 10;) = 0.1937.$$

---

## Evaluation on a test set

- Suppose you trained a classifier  $A \Rightarrow$  on 10 examples,  $\hat{\epsilon}_A = 0.1$
- Another classifier  $B \Rightarrow$  test error  $\hat{\epsilon}_B = 0.2$
- Which one is better? Naive approach:  $A$  since  $\hat{\epsilon}_A < \hat{\epsilon}_B$ .
- What if in reality, the expected risk is  $\epsilon_A = 0.3$  and  $\epsilon_B = 0.015$ ?
  - Probability of observing  $\hat{\epsilon}_A$  is

$$\text{Binomial}_{0.3}(1, 10;) = 0.1211,$$

and probability of observing  $\hat{\epsilon}_B$  is

$$\text{Binomial}_{0.1}(2, 10;) = 0.1937.$$

- So, we could get the observed results with reasonably high probability even if in fact  $A$  had expected risk twice that of  $B$ .

---

# Hypothesis testing

- The *null hypothesis*  $H_0 : \epsilon_A = \epsilon_B$ .
- The *alternative hypothesis*:  $H_1 : \epsilon_A < \epsilon_B$ .
  - We want to *reject*  $H_0$  (in favor of  $H_1$ ).
  - Note: if we can reject  $H_0$  we can also reject  $H' : \epsilon_A < \epsilon_B$ .
- Many, many tests exist in statistics.
- Two types of error in a hypothesis test:
  - Type I: reject  $H_0$  when  $H_0$  is true.
  - Type II: fail to reject  $H_0$  when  $H_1$  is true.

---

## $p$ -value

$$H_0 : \epsilon_A = \epsilon_B \quad H_1 : \epsilon_A > \epsilon_B.$$

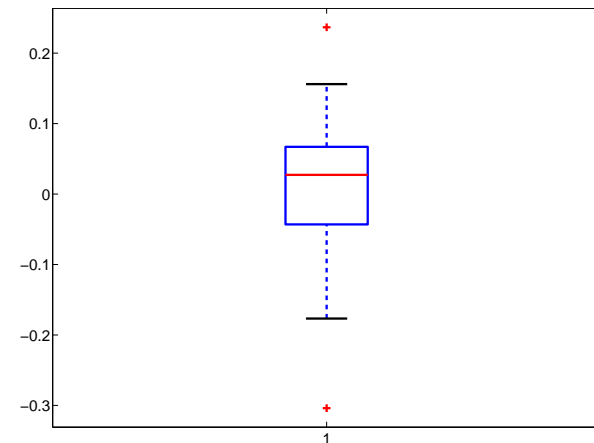
- Typically, a test is based on a *statistic*  $\hat{\theta}$  (function of the data).
  - e.g.,  $\hat{\theta} = \textit{epsilon}_A - \hat{\epsilon}_B$ .
- The  $p$ -value of the test on our data: the probability of observing a value of the test statistic *the same or more extreme* than that actually observed, *under the null-hypothesis*  $H_0$ .
- Standard interpretation:

$p < .01$	very strong evidence against $H_0$ ;
$.01 < p < .05$	strong evidence against $H_0$ ;
$.05 < p < .1$	weak evidence against $H_0$ ;
$p > .1$	no evidence.

## Informal assessment of results

- Suppose you have (test) regression errors  $e_1, \dots, e_N$  with method  $A$  and  $e'_1, \dots, e'_N$  with method  $B$ .
- This calls for a *paired* test (samples are not independent).
- An obvious statistic: the mean difference  $\frac{1}{N} \sum_i (e_i - e'_i)$

- Matlab's `boxplot`:



---

# Significance and importance

- Note that the test typically depends on  $N$ .
  - If  $N = 3$ ,  $\hat{\epsilon}_A = 1/3$  and  $\hat{\epsilon}_B = 2/3$ , not significant.
  - If  $N = 10^6$  and  $\hat{\epsilon}_A = 0.03$  and  $\hat{\epsilon}_B = 0.06$ , probably significant.
- Note: statistically significant  $\neq$  important!

---

# What have we not covered?

- Inference in graphical models
  - Exact: local message passing algorithm
  - Approximate: sampling, loopy belief propagation.
- Undirected models: Markov random fields
- Example: image models

---

# Structure learning

- We have assumed that the structure (edges) of the GM is given.
- We can learn it from data
  - A model selection task.
  - The best explanation: fully connected graph.
  - Need to penalize it by model complexity.
- (Non-linear) manifold learning
  - PCA recovers an “interesting” linear subspace of the data.
  - Many methods target non-linear subspaces.

---

# Semi-supervised learning

- Small labeled data set  $\sim p(\mathbf{x}, y) = p(\mathbf{x})p(y | \mathbf{x})$ ;
- Large set of unlabeled data  $\sim p(\mathbf{x})$ .
- We can use the unlabeled data to improve the model/estimates
  - Estimate density, and use the result to assign weights to labeled examples.
  - *Transduction*: predict the labels for the unlabeled data, and re-train the classifier pretending these are correct.

---

# Active learning

- We are allowed to query the label of unlabeled examples.
- Labeling is expensive.
  - Recall: in linear regression,

$$\hat{\mathbf{w}} \sim \mathcal{N}(\mathbf{w}; \mathbf{w}^*, \sigma^2(\mathbf{X}^T \mathbf{X})^{-1})$$

- Basic idea: query examples whose label will contribute most to your ability to predict future labels.

---

# Online learning

- We observe examples in order, and start learning right away
- With each example (or small batch of examples) need to update the model
  - Often need to make predictions quickly!
- Applications:
  - Financial time series prediction
  - Adaptive systems
  - Robot exploration of environment

---

# Reinforcement learning

- “Reinforcement learning is learning what to do—how to map situations to actions—so as to maximize a numerical reward signal.” [Sutton & Barto]
- Main elements:
  - *Actions* that can be taken
  - *Policy*: mapping from state of the environment to action.
  - *Reward* function: mapping state-action pairs to value.
- Objective: through trial and error, learn a policy that will maximize expected reward in the long run.
- Examples: many. E.g., inverted helicopter.

---

# Theory questions

- What is learnable with a particular family of classifiers?

---

# Theory questions

- What is learnable with a particular family of classifiers?
- Probably Approximately Correct (PAC) framework:
  - We select from a set  $\mathcal{H}$  a hypothesis  $h^*$  that achieves zero training error.
  - How large should  $N$  be so that with probability at least  $1 - \delta$ , the expected risk of  $h^*$  is no more than  $\epsilon$ ?