FRIEDRICH-ALEXANDER-
UNIVERSITÄT
ERLANGEN-NÜRNBERG

SCHOOL OF ENGINEERING

Lecture Pattern Analysis

Part 17: Short Recap and Remarks on the Exam

Christian Riess
IT Security Infrastructures Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg
July 4, 2022

Introduction and High-Level Review

- We finished Parts 1 and 2 of Pattern Analysis
  - Part 1 focuses on representations of a sample space
  - Part 2 focuses on simplifications via clustering and manifold learning
- High-level remarks on some (potentially implicit) aspects are listed below

1. Besides the actual algorithms, PA also has a “latent space” of topics
   - Local operators in the feature space
   - Tradeoffs on model complexity and flexibility (e.g., fixed kernel neighborhoods vs. data-driven random forest neighborhoods)
   - Model selection, this year with some extensions for GMMs
2. Tools/tricks can oftentimes be re-used for different tasks:
   - Space partitioning: Random Forest splits can be used in classification, regression, DE, ML (btw., the same is true for kernels and the k-NN operator)
   - Gibbs Sampling is used for GMM fitting and also for MRF inference

High-Level Review (Continued)

3. Different optimization criteria or algorithm variants can lead to different results:
   - Kernel density estimates can be zero somewhere, K-NN density estimates not
   - The clusters from k-means, mean shift, GMMs can have very different shapes
   - The gap statistics is a good match for k-means clusters, but not for mean shift
   - PCA/MDA/ISOMAP projections preserve long distances, LE preserves local neighborhoods. LE is more robust on curved manifolds
4. Algorithm assumptions are important
   - Kernel density estimation assumes a number of samples in the kernel window
   - Random Forest training must decorrelate the trees, at least to some extent
   - Individual trees in a random forest must perform better than random guessing (50% in a 2-class problem)
5. Computational requirements (space/time) are important
   - Kernel density estimation must store and lookup all samples for a query, or pre-compute the whole d-dim. density
   - Random Forests variants are oftentimes more efficient (and also more expressive), even more since they are trivially parallelized

Hints on the Exam

- 60 minutes, 60 points, just a pen (no books, cheat sheets, ...)
- Most questions require 1–3 sentences as answer
- Few questions require a sketch, very few are multiple choice
- Questions will be a combination of three levels of mental productivity:
  1. Reproduction Questions, for example
     - Write down the objective function for calculating the mean shift vector
     - State the algorithmic steps of the ISOMAP algorithm
     - Name 3 options for randomization in Random Forest training
  2. Explanation Questions, for example
     - How does a Random Forest achieve a smooth classification boundary?
     - What complicates working in high-dimensional spaces?
  3. Comparison / Analysis Questions, for example
     - Was the density in Fig. X created from a box kernel or Gaussian kernel? Why?
     - Sketch a sample distribution where Laplacian Eigenmaps with kernel-based affinities might fail, but Manifold Forests might work

Hints on the Preparation

- It may pay to practice already during the preparation short answers to some questions
- There are two sources of preparation material online:
  - The PA 2022 class material (our studOn class):
    This is the reference for the exam, including everything that is on studOn
  - The PA 2021 class material (on video.fau.de):
    Short videos; large overlap with 2022, but please check for differences
- The exam will not require you to write code
- The exam will not have questions on content that only occurs in the supplemental literature (Bishop, Hastie/Tibshirani/Friedman) but not in the lecture/exercises/joint meetings
- Many learner types benefit from learning groups
  The virus tries to screw this up to the greatest extent possible, but please try to reach out to colleagues

FRIEDRICH-ALEXANDER-
UNIVERSITÄT
ERLANGEN-NÜRNBERG

SCHOOL OF ENGINEERING

Lecture Pattern Analysis

Part 18: Introduction to Probabilistic Graphical Models

Christian Riess

IT Security Infrastructures Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg

July 5, 2022

Introduction

- We will now look deeper into relationships between random variables
- Probabilistic graphical models constitute the third part of PA[1]
- We are already familiar with probabilistic models:
  - Probabilistic models represent data properties in a probabilistic formulation
  - We usually start with a joint PDF of all random variables $p(\mathbf{x}_1, \ldots, \mathbf{x}_N)$
  - Factorization via the product rule creates some smaller terms:

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = p(\mathbf{x}_N | \mathbf{x}_1, \ldots, \mathbf{x}_{N-1}) \cdot \ldots \cdot p(\mathbf{x}_2 | \mathbf{x}_1) \cdot p(\mathbf{x}_1) \tag{1}$$

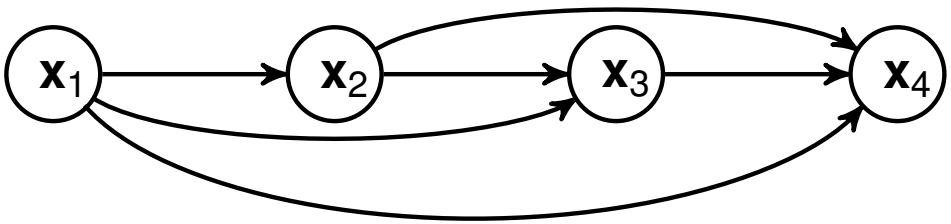
  - The order of factorization is arbitrary. For example, this is also correct:

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = p(\mathbf{x}_3 | \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_4, \ldots, \mathbf{x}_N) \cdot \ldots \cdot p(\mathbf{x}_2 | \mathbf{x}_1) \cdot p(\mathbf{x}_1) \tag{2}$$

[1] Literature reference for this video is Bishop Sec. 8 and Sec. 8.1 (just the first 4 pages, i.e., without Sec. 8.1.1 until 8.1.4)

Graphical Models

- A graph representation aims to improve the understanding of these relationships, and the reasoning (inference) on the variables
  - Directed edges represent **conditional probabilities** (the general class of such graphs is called Bayesian networks).
    We will study Hidden Markov Models (HMMs) as an important special case
  - Undirected edges represent joint probabilities. We will study Markov Random Fields (MRFs) as an important special case.
- Example graph for $p(\mathbf{x}_4|\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \cdot p(\mathbf{x}_3|\mathbf{x}_1, \mathbf{x}_2) \cdot p(\mathbf{x}_2|\mathbf{x}_1) \cdot p(\mathbf{x}_1)$:

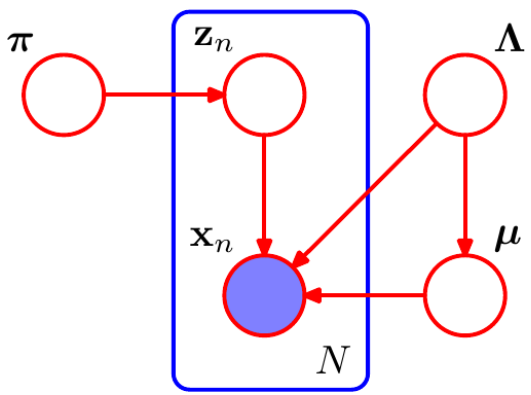


  where an edge expresses a conditional probability. In particular,
  - $\mathbf{x}_1$ conditions all variables $\rightarrow$ 3 outgoing edges
  - $\mathbf{x}_4$ depends on all other variables $\rightarrow$ 3 incoming edges
  - $\mathbf{x}_2$ depends on $\mathbf{x}_1$ and conditions $\mathbf{x}_3$, $\mathbf{x}_4$ $\rightarrow$ 1 incoming edge, 2 outgoing edges

Graphical Models and Probabilistic Inference

- As mentioned for the model selection on GMMs, inference on a full probabilistic model is oftentimes infeasible
- Assumptions on variable independence can solve this issue
- The associated graphical model has no edge between independent variables
- For example, the graphical model for GMM fitting via Bishops variational approximation is

Figure 10.5 Directed acyclic graph representing the Bayesian mixture of Gaussians model, in which the box (plate) denotes a set of N i.i.d. observations. Here $\boldsymbol{\mu}$ denotes $\{\boldsymbol{\mu}_k\}$ and $\boldsymbol{\Lambda}$ denotes $\{\boldsymbol{\Lambda}_k\}$.



- In this Chapter, we will investigate independence assumptions to obtain tractable models for sequences (HMMs) and label assignment tasks (MRFs)

FRIEDRICH-ALEXANDER-
UNIVERSITÄT
ERLANGEN-NÜRNBERG

SCHOOL OF ENGINEERING

Lecture Pattern Analysis

Part 19: Conditional Independence

Christian Riess
IT Security Infrastructures Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg
July 5, 2022

Overview

- Variable independence enables factorization, and hence training and inference
- Conditional independence is an important special case[1]
- If $p(a|b,c) = p(a|c)$, then a is **conditionally independent** of b given c,

$$a \perp\!\!\!\perp b|c \tag{1}$$

- Conditional independence may occur in more complex expressions, e.g.,

$$p(a,b|c) = p(a|b,c)p(b|c) \tag{2}$$

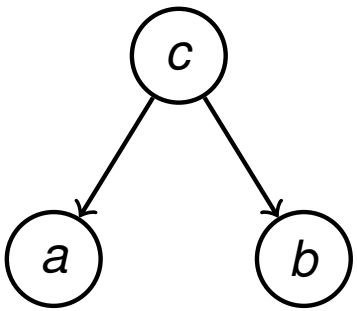
$$\stackrel{!}{=} p(a|c)p(b|c) \tag{3}$$

- Conditional independence is somewhat difficult on directed graphs, simpler on undirected graphs
- The **Markov blanket** makes inference on undirected graphs straightforward, which we will use for the upcoming Markov Random Fields

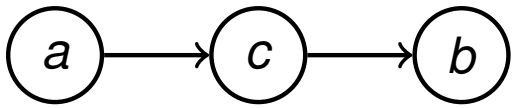
[1] The reference for this chapter is Bishop Sec. 8.2 and Sec. 8.3.1

Conditional Independence on Directed Graphs

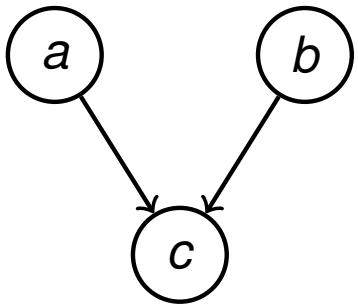
- We study three directed subgraphs that factorize $p(a, b, c)$:



$p(c)p(a, b|c)$

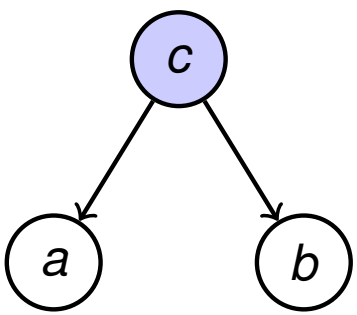


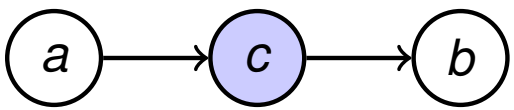
$p(a)p(c|a)p(b|c)$

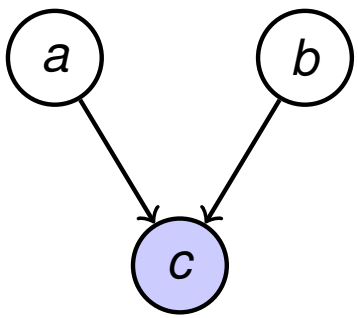


$p(a)p(b)p(c|a, b)$

- Shading indicates that a variable is observed:

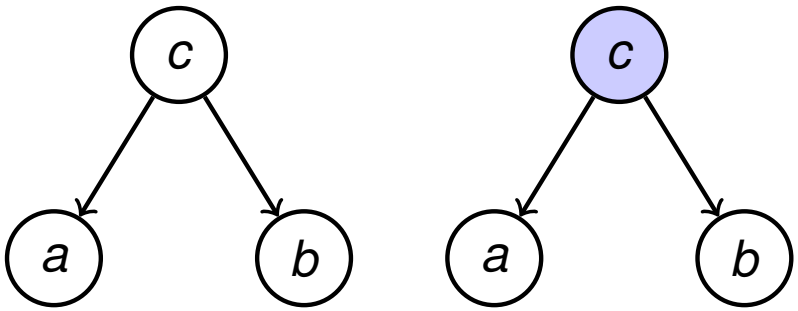






- If c is observed, a and b become cond. **independent** in the first two graphs
- In the third graph, a and b become conditionally **dependent** if c is observed

Conditional Independence on the Tail-to-Tail Graph



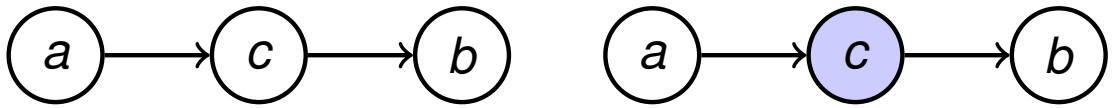
- If c is unobserved: $a \not\perp\!\!\!\perp b \mid c$, since $p(a,b) \neq p(a)p(b)$:
  $p(a,b)$ is obtained by marginalizing over the possible values of c,

$$p(a,b) = \sum_c p(a|c)p(b|c)p(c) \neq p(a)p(b) \ , \tag{4}$$

- If c is observed, then $a \perp\!\!\!\perp b \mid c$:

$$p(a,b|c) = \frac{p(a,b,c)}{p(c)} = \frac{p(c)p(a|c)p(b|c)}{p(c)} = p(a|c)p(b|c) \tag{5}$$

Conditional Independence on the Head-to-Tail Graph



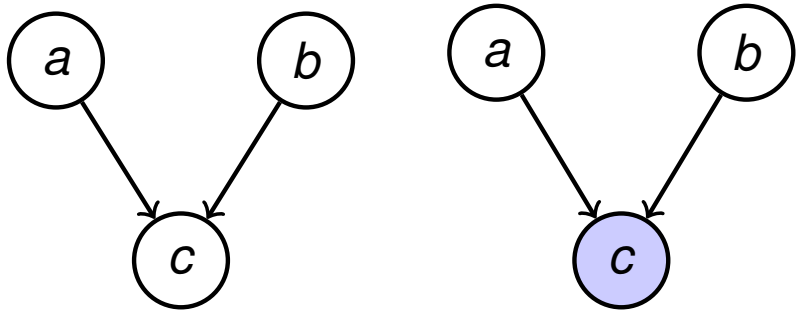
- If c is unobserved: $a \not\perp\!\!\!\perp b \mid c$, since $p(a,b) \neq p(a)p(b)$:

$$p(a,b) = \sum_c p(a)p(c|a)p(b|c) = p(a)p(b|a) \neq p(a)p(b) \ , \tag{6}$$

- If c is observed, then $a \perp\!\!\!\perp b \mid c$:

$$\begin{aligned} p(a,b|c) &= \frac{p(a,b,c)}{p(c)} = \frac{p(a)p(c|a)p(b|c)}{p(c)} = \frac{p(c)p(a|c)p(b|c)}{p(c)} \\ &= p(a|c)p(b|c) \end{aligned} \tag{7}$$

Conditional Independence on the Head-to-Head Graph



- If c is unobserved: $a \perp\!\!\!\perp b \mid c$, since $p(a,b) = p(a)p(b)$:

$$p(a,b) = \sum_c p(a)p(b)p(c|a,b) = p(a)p(b) \tag{8}$$

- If c is observed, then $a \not\perp\!\!\!\perp b \mid c$:

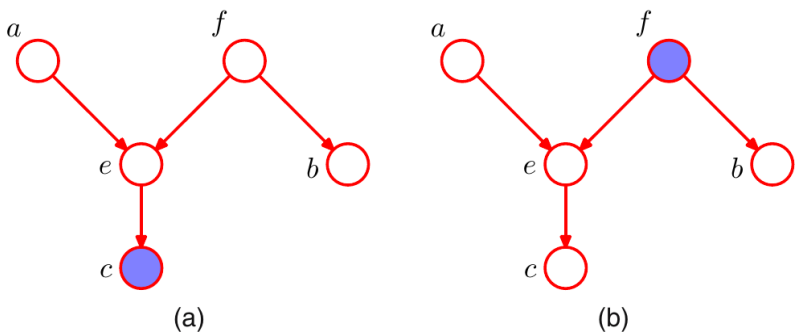
$$p(a,b|c) = \frac{p(a,b,c)}{p(c)} = \frac{p(a)p(b)p(c|a,b)}{p(c)} \neq p(a|c)p(b|c) \tag{9}$$

- Bishop Sec. 8.2.1 provides a numerical example to further illustrate this counter-intuitive case

D-Separation of Variables in Directed Graphs

- On more complex directed graphs, D-separation indicates whether for sets of nodes A, B, C the conditional independence $A \perp\!\!\!\perp B \mid C$ holds
- Variables in C are observed
- Consider all paths between A and B
- A path is **blocked** if it contains a node where
  - the node is in C and its arrows are tail-to-tail or head-to-tail
  - the node and its decendants are not in C and its arrows are head-to-head
- If all paths are blocked, A is d-separated from B by C, i.e., $A \perp\!\!\!\perp B \mid C$

Figure 8.22 Illustration of the concept of d-separation. See the text for details.

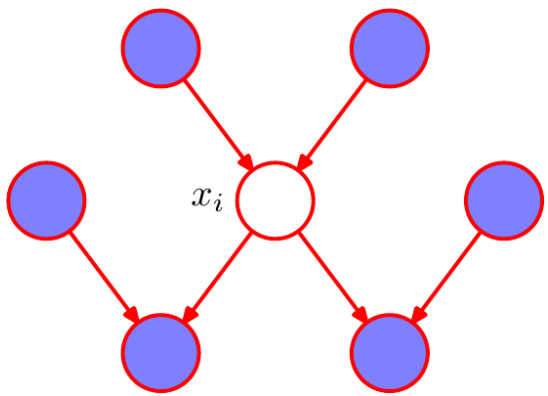


- Example: Left: a and b are dependent. Right: a and b are independent

Markov Blanket on Directed Graphs

- The minimal set of nodes that isolates a node from the rest of the graph is called a **Markov blanket**
- It includes the immediate parents, immediate children, and co-parents of immediate children:

Figure 8.26 The Markov blanket of a node $\mathbf{x}_i$ comprises the set of parents, children and co-parents of the node. It has the property that the conditional distribution of $\mathbf{x}_i$, conditioned on all the remaining variables in the graph, is dependent only on the variables in the Markov blanket.

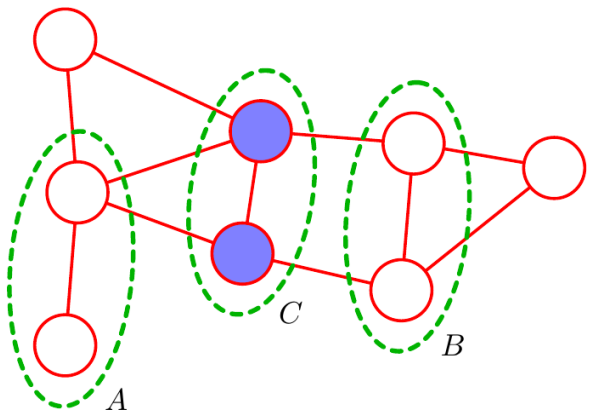


- In HMM training, we (luckily) have a much simpler dependency structure:
  - The iterative parameter updates assume that the previous and future states are tentatively fixed
  - Hence, we can use α and β as aggregated path probabilities to update $\boldsymbol{\pi}$, $\mathbf{A}$, $\mathbf{B}$

Conditional Independence and Markov Blanket on Undirected Graphs

- Conditional independence on undirected graphs is simpler: just separate *A* and *B* by observed nodes

Figure 8.27 An example of an undirected graph in which every path from any node in set A to any node in set B passes through at least one node in set C. Consequently the conditional independence property $A \perp\!\!\!\perp B \mid C$ holds for any probability distribution described by this graph.



- Analogously, the Markov blanket just includes the set of neighbors of a node:

Figure 8.28 For an undirected graph, the Markov blanket of a node x_i consists of the set of neighbouring nodes. It has the property that the conditional distribution of x_i, conditioned on all the remaining variables in the graph, is dependent only on the variables in the Markov blanket.