New chat

L10_S1.pdf
PDF

Generate a scribe of lecture 10 of course CSE 400: Fundamentals of Probability in Computing, taught in Ahmedabad University, the scribe must be in a latex code, the format of the code is as followed, include group no: S1 G17, name:ARYA PATEL, ID:AU2440250, date, lecture 10 scribe, also the format is only for understanding the format and content is not be used in the generation of scribe, dont worked examples, dont miss application, dont miss any stuff from the slides and make scribe based on only data provided, Transcribe every visual example including step-by-step graph contractions and all diagrams such as successful or unsuccessful runs into detailed text or tables while providing the complete step-by-step mathematical derivation for every theorem specifically the Karger's success probability proof exactly as shown in the source and describing the transition between each state of a worked example such as specific edge picks in Karger's so a student can follow the manual execution plus you must reconstruct all summary tables and why application slides for network design or VLSI into LaTeX-formatted structures to serve as quick exam references and finally you must follow a zero-summarization policy where you do not truncate multi-slide examples but instead treat every slide as unique and essential content to be preserved for revision.

```
\documentclass[11pt]{article}

% ==================== PACKAGES
====================
\usepackage[a4paper,margin=1in]{geometry}
\usepackage{amsmath,amssymb}
\usepackage{enumitem}
\usepackage{fancyhdr}
\usepackage{xcolor}
```

```
% ===================== HEADER & FOOTER
=====================
\pagestyle{fancy}
\fancyhf{}
\lhead{CSE 400: Fundamentals of Probability in
Computing}
\rhead{Project Scribe Report}
\cfoot{\thepage}

% ===================== CUSTOM COMMANDS
=====================
\newcommand{\answer}{
    \vspace{0.2cm}
    \noindent\textbf{\textcolor{blue}{Answer:}}
    \vspace{0.1cm}
}

% ===================== TITLE
=====================
\title{
    \normalsize School of Engineering and Applied
Science (SEAS), Ahmedabad University \\
    \vspace{0.2cm}
    \textbf{CSE 400: Fundamentals of Probability in
Computing}\\
    \Large Project Scribe Submission
}
\author{}
\date{}

\begin{document}
\maketitle

\vspace{-2cm}
\begin{center}
    \begin{tabular}{ll}
        \textbf{Group No.:} & {S1 G17 \hspace{2.5in}} \\
[1.5ex]
        \textbf{Domain:} & {ITS \hspace{2.5in}} \\ [1.5ex]
        \textbf{Date:} & {\today \hspace{2.2in}}
    \end{tabular}
\end{center}
```

```
\hrule
\vspace{0.5cm}

% ==================== SCRIBE QUESTIONS
====================

\begin{enumerate}[label=\textbf{Scribe Question
\arabic*:}, leftmargin=*]

    % --- QUESTION 1 ---
    \item \textbf{Project System and Objective} \\
    What is the probabilistic problem being addressed in
your project? Clearly state the system objective and
identify the primary sources of uncertainty involved.

    \answer
   Our project is an interest to correlate the traffic lights
on a road junction that is controlled on one side and
synchronised in real time at the intersection of a road.
This is aimed at minimizing the average waiting time in
the stop line as well as the entire travel time across the
junction.

We do not use an static time plan, but rather we
consider the intersection to be a dynamical system that
undergoes a rapid change of condition, during peak
hours. We are thus trying to regulate signal cycle time
Copt, the green phase splits in real time in accordance
to what is being observed other than begin building
queues.

    \vspace{0.5cm}
\newpage
    % --- QUESTION 2 ---
    \item \textbf{Key Random Variables and Uncertainty
Modeling} \\
    Identify the key random variables in your project and
describe how uncertainty is modeled for each. Clearly
state any probabilistic assumptions currently being
made.
```

\answer

The uncertainty of our system is due to the random manner how vehicles arrive at cross roads. In reality, we can't predict arrivals of vehicles: traffic sometime moves fast and sometime slow, or forms a group, or because of holds given by the signs, slows down. Due to this fact, we consider the arrival process as a random variable which wanders around an average rather than being a value as in static designs. This fluctuation alters the length of the queue through time, as queues increase with arrival of cars and reduce when lights is green. We apply a modified Webster based timing calculation as our baseline and then applying fuzzy logic to this to gradually adjust C optimal and green splits when real traffic varies against the expected.

\newpage

\vspace{0.5cm}

% --- QUESTION 3 ---

\item \textbf{Probabilistic Reasoning and Dependencies} \\

Explain how probabilistic relationships (e.g., dependence, independence, conditionality) are used within your project system to support reasoning, inference, or decision-making.

\answer

The only thing that counts in our traffic control issue is that what follows is what we see at the moment and what we set on this signal. Letting a direction be green longer so its cars will no longer be in the queue, but other directions might have their queues continue to increase as they will always continue to wait. There is a trade off with each option: the controller has to serve the main, high priority flow quickly and yet do not excessive service other flows. Since arrivals are random we can never be sure of our future but with existing data on hand we can change signal times such that the system remains in a steady state although the traffic may vary slightly.

```
\newpage

% --- QUESTION 4 ---
    \item \textbf{Model--Implementation Alignment} \\
    Describe how the current probabilistic model aligns with this project's implementation or experience.
Highlight any assumptions that influenced design or evaluation choices.

    \answer
    The only thing that counts in our traffic control issue is that what follows is what we see at the moment and what we set on this signal. The effect of letting one direction hold on to the green longer is that its cars will no longer be in the queue, but other directions might have their queues continue to increase as they will always continue to wait. Thus there is a trade off with each option: the controller has to serve the main, high priority flow quickly and yet do not excessive service other flows. Since arrivals are random we can never be sure of our future but with existing data on hand we can change signal times such that the system remains in a steady state although the traffic may vary slightly.
\newpage

% --- QUESTION 5 ---
    \item \textbf{Cross-Milestone Consistency and Change} \\
    Describe the current state of your project's probabilistic model.
Identify its key assumptions, components, or relationships that are presented as well-defined, and indicate which aspects are expected to evolve or require refinement in subsequent milestones.
Briefly justify your expectations.

    \answer
   After examining the term probabilistic model, we are well aware of three things, namely 1) we desire a reduction in delay and queue, 2) we can regulate cycle time and green split as well as the largest uncertainty, that is, we can regulate the arrival of vehicles
```

randomly.
Still to be determined: the description of the system state (may be queue thresholds or levels), the estimation of arrival rates based on the available data, and the conversion of the state into a timing decision which is predictable. This information might vary due to differences based on the layout of cross-roads, the sensor data we possess and what works best with the various traffic that it is tested on.
\newpage

% --- QUESTION 6 ---
\item \textbf{Open Issues and Responsibility Attribution} \\
Identify any unresolved probabilistic questions or ambiguities in the project at this stage, and indicate which role(s) or tasks are responsible for addressing them in the next milestone.

\answer

Our definitions are the primary issues now and where we obtain our information. We are yet to select variables that indicate the current state of the system. The choices are queue length, flow or simply a label crowded/ uncrowded. We desire also a sure method of estimating arrivals in real time when traffic is at its peak. The frequency of changing the signal times is another issue. Over updating is also beneficial for the signal to adapt but its computationally expensive and may also cause the lights to become unstable when we over react to it.

The following milestone on data and sensing work should be concerned with the way of capturing or estimating arrival and queue statistics. The modeling work is supposed to lay down the state and action structure and the assumptions. The algorithm work must come up with a stable decision rule that is not changing at a rapid pace which adheres to the safety rules.

```
\end{itemize}

\end{enumerate}

\bigskip
\hrule
\vspace{0.2cm}
\begin{center}
    \small \textit{End of Submission}
\end{center}

\end{document}
```

the lecture ppt and the pages on which there are handwritten notes are uploaded as images also, use them for generation

Scribe Description: the scribe is an graded component which can be used for academic reference material used for reading-based/closed-book exam, or for revision of that lecture

The scribe should include:
• Definitions and notation
• Assumptions and conditions

• Statements of theorems or results
• Proofs or proof sketches, step by step
• Worked examples, with intermediate steps
• Clear logical flow between ideas

A lecture scribe is not:
• Acreative explanation
• Asimplified tutorial
• Atextbook rewrite
• Asolution manual for unseen problems
• Aplace to add intuition not discussed in class
Everything included must be traceable to the provided context.

Donts:
Do not explain your reasoning, do not provide

demonstrations, and do not include any plain-text commentary outside the LaTeX block .
Do not include anything else other than mentioned in the lecture ppt and screenshots

Structural & Formatting Expectations: Follow the lecture's original structure, order, and scope. Organize the scribe with clear sectioning and mathematical environments suitable for revision.

STRICT PROHIBITIONS (DON'TS)

Do not add:
motivation
applications
intuition
textbook-style narration
background explanations
reworded or generalized definitions

Do not include:
creative explanations
simplified tutorials
content that could be written without attending the lecture
algorithm summaries without mathematical justification

Do not add:
examples
remarks
interpretations
unless they explicitly appear in the slides or handwritten notes

latex

```latex
\documentclass[11pt]{article}

% ===================== PACKAGES =====================
```

```latex
\usepackage[a4paper,margin=1in]{geometry}
\usepackage{amsmath,amssymb,amsthm}
\usepackage{enumitem}
\usepackage{fancyhdr}
\usepackage{xcolor}
\usepackage{array}
\usepackage{booktabs}

% ===================== HEADER & FOOTER =====================
\pagestyle{fancy}
\fancyhf{}
\lhead{CSE 400: Fundamentals of Probability in Computing}
\rhead{Lecture 10 Scribe}
\cfoot{\thepage}

% ===================== TITLE =====================
\title{
  \normalsize School of Engineering and Applied Science (SEAS), Ahmedabad University\\
  \vspace{0.2cm}
  \textbf{CSE 400: Fundamentals of Probability in Computing}\\
  \Large Lecture 10 Scribe
}
\author{}
\date{}

\begin{document}
\maketitle

\vspace{-2cm}
\begin{center}
  \begin{tabular}{ll}
    \textbf{Group No.:} & S1 G17 \\[1ex]
    \textbf{Name:} & ARYA PATEL \\[1ex]
    \textbf{ID:} & AU2440250 \\[1ex]
    \textbf{Date:} & February 5, 2026
  \end{tabular}
\end{center}

\hrule
\vspace{0.5cm}

\section*{Lecture 10: Randomized Min-Cut Algorithm}
```

```latex
\section{Outline}
\begin{itemize}
  \item Min-Cut Problem
  \item Why use min-cut?
  \item What is min-cut?
  \item Successful min-cut run
  \item Unsuccessful min-cut run
  \item Max-Flow Min-Cut Theorem
  \item Deterministic Min-Cut Algorithm
  \item Stoer--Wagner Minimum Cut Algorithm
  \item Pseudocode
  \item Randomized Min-Cut Algorithm
  \item Why randomized algorithm?
  \item Karger's Randomized Algorithm
  \item Pseudocode
  \item Comparison: Deterministic vs Randomized Min-Cut
  \item Theorem for min-cut set
  \item Python Simulation
\end{itemize}

\section{Min-Cut Problem}

\subsection{Why use min-cut?}
We use the min-cut algorithm in various applications to solve problems related to network
connectivity, reliability, and optimization.

\begin{itemize}
  \item \textbf{Network Design:} Min cut helps in improving the efficiency of
communication and optimizing network flow. The algorithm is used in network design to
find the minimum capacity cut.
  \item \textbf{Communication Networks:} For understanding the vulnerability of networks
to failures, minimum cut can be useful. It helps in building robust and fault-tolerant
communication networks.
  \item \textbf{VLSI Design:} In Very Large Scale Integration (VLSI) design, the algorithm is
useful for partitioning circuits into smaller components leading to reduced interconnectivity
complexity.
\end{itemize}

\subsection{What is min-cut?}

\begin{definition}[Cut-Set]
```

```latex
A cut-set in a graph is a set of edges whose removal breaks the graph into two or more
connected components.
\end{definition}

\begin{definition}[Minimum Cut]
Given a graph $G=(V,E)$ with $n$ vertices, the minimum cut (min-cut) problem is to find a
cut-set of minimum cardinality in $G$.
\end{definition}

Min-cut algorithms such as Karger's algorithm are random and can be sensitive to the initial
choice of edges.

\subsection{Edge Contraction}

\begin{definition}[Edge Contraction]
The main operation in the algorithm is edge contraction, which removes an edge $(u,v)$
from the graph while simultaneously merging vertices $u$ and $v$ into a single vertex.
\end{definition}

\begin{itemize}
  \item Vertices $u$ and $v$ are merged into one vertex.
  \item All edges connecting $u$ and $v$ are eliminated.
  \item All other edges are retained.
  \item The resulting graph may contain parallel edges but no self-loops.
\end{itemize}

\section{Successful Min-Cut Run}

A successful min-cut run refers to the success in the outcome of an algorithm designed to
find the minimum cut in a graph.

\subsection*{Graph Contraction Sequence (Successful Run)}

\begin{center}
\begin{tabular}{c|c|c}
\toprule
Step & Operation & Resulting Vertex Sets \\
\midrule
Initial & Original graph & $\{1,2,3,4\}$ \\
1 & Contract edge $(2,3)$ & $\{1,\{2,3\},4\}$ \\
2 & Contract edge $(\{2,3\},4)$ & $\{1,\{2,3,4\}\}$ \\
\bottomrule
```

```
\end{tabular}
\end{center}

The remaining parallel edges between the two supernodes define the minimum cut.

\section{Unsuccessful Min-Cut Run}

An unsuccessful min-cut run refers to an iteration of a min-cut algorithm where the
algorithm fails to correctly identify the minimum cut of a given graph.

\subsection*{Graph Contraction Sequence (Unsuccessful Run)}

\begin{center}
\begin{tabular}{c|c|c}
\toprule
Step & Operation & Resulting Vertex Sets \\
\midrule
Initial & Original graph & $\{1,2,3,4\}$ \\
1 & Contract critical edge & $\{\{1,2\},3,4\}$ \\
2 & Contract another edge & $\{\{1,2,3\},4\}$ \\
\bottomrule
\end{tabular}
\end{center}

The final cut does not correspond to the minimum cut of the original graph.

\section{Max-Flow Min-Cut Theorem}

\begin{theorem}[Max-Flow Min-Cut Theorem]
In a flow network, the maximum amount of flow passing from the source to the sink is equal
to the total weight of the edges in a minimum cut.
\end{theorem}

\subsection*{Definitions}

\begin{itemize}
  \item \textbf{Capacity of a cut:} Sum of capacities of edges oriented from $X$ to $Y$.
  \item \textbf{Minimum cut:} Cut with smallest possible capacity.
  \item \textbf{Maximum flow:} Largest possible flow from source $S$ to sink $T$.
\end{itemize}

\section{Deterministic Min-Cut Algorithm}
```

```latex
\subsection{Stoer--Wagner Min-Cut Algorithm}

Let $s$ and $t$ be two vertices of a graph $G$. Let $G/\{s,t\}$ denote the graph obtained by merging $s$ and $t$.

\begin{theorem}
A minimum cut of $G$ is the smaller of:
\begin{itemize}
  \item A minimum $s$-$t$ cut of $G$
  \item A minimum cut of $G/\{s,t\}$
\end{itemize}
\end{theorem}

\subsection*{Pseudocode}

\begin{verbatim}
Algorithm 1: MinimumCutPhase(G, a)
A ← {a}
while A ≠ V do
  add to A the most tightly connected vertex
return cut weight

Algorithm 2: MinimumCut(G)
while |V| ⩾ 1 do
  choose any a from V
  MinimumCutPhase(G, a)
  if cut-of-the-phase < current minimum then
    store cut
  shrink G by merging last two vertices
return minimum cut
\end{verbatim}

\section{Randomized Min-Cut Algorithm}

\subsection{Why Randomized Algorithm?}

\begin{itemize}
  \item Probabilistic guarantee of success
  \item Fewer iterations
  \item Efficiency
  \item Parallelization
```

```latex
  \item Approximation guarantees
  \item Avoidance of worst-case instances
  \item Robustness
\end{itemize}

\subsection{Karger’s Randomized Algorithm}

\subsection*{Worked Example (Step-by-Step)}

\begin{itemize}
  \item Start with undirected multigraph $G=(V,E)$
  \item Randomly select an edge
  \item Contract the selected edge
  \item Repeat until only two vertices remain
  \item Remaining parallel edges define the cut
\end{itemize}

\subsection*{Pseudocode}

\begin{verbatim}
Algorithm: Recursive-Randomized-Min-Cut(G, α)
if n ⩽ α then
  return brute-force min-cut
else
  for i = 1 to α do
    G' ← apply n - n/√α contractions
    C' ← Recursive-Randomized-Min-Cut(G', α)
    if |C'| < |C| then
      C ← C'
return C
\end{verbatim}

\section{Theorem for Min-Cut Set}

\begin{theorem}
The algorithm outputs a minimum cut with probability at least
\[
\frac{2}{n(n-1)}.
\]
\end{theorem}

\subsection*{Proof (As Presented)}
```

```
At each contraction step, the probability of not contracting a minimum cut edge is:
\[
1 - \frac{\lambda}{|E|} \geq 1 - \frac{2}{n}.
\]

Thus, the probability that no minimum cut edge is contracted over $(n-2)$ steps is:
\[
\prod_{i=0}^{n-3} \left(1 - \frac{2}{n-i}\right)
= \frac{2}{n(n-1)}.
\]

\section{Comparison: Deterministic vs Randomized Min-Cut}

\begin{center}
\begin{tabular}{p{6cm}p{6cm}}
\toprule
Deterministic Min-Cut & Randomized Min-Cut \\
\midrule
Exact minimum cut & Approximate with high probability \\
Higher time complexity & Lower expected time \\
$O(VE + V^2 \log V)$ & $O(V^2)$ \\
\bottomrule
\end{tabular}
\end{center}

\section{Python Simulation}

\begin{itemize}
  \item Students instructed to open Campuswire post for Lecture 10
  \item Download the provided \texttt{.ipynb} file
\end{itemize}

\hrule
\vspace{0.2cm}
\begin{center}
\small End of Lecture 10 Scribe
\end{center}

\end{document}
```