School of Engineering and Applied Science (SEAS), Ahmedabad University

CSE400: Fundamentals of Probability in Computing

Group Name: *S*2 *Bio* 1
Team Members:

- Aashi Shah (AU2340043)
- Manushree Patel (AU2340014)
- Twisha Patel (AU2340056)
- Kavish Vacheta (AU2340190)
- Nihar Kikani (AU2340046)

I. Background and Motivation

- **Project Title:** Probabilistic Analysis of Fruit Fly Social Interactions
- **Aim:** The aim is to model *Drosophila melanogaster* behavior using a Dynamic Bayesian Network to identify patterns and predict its behavior.

A. Background

- **Model Organism**: The fruit fly, Drosophila melanogaster, is a popular choice for behavioral studies due to its relatively simple nervous system, short lifespan, and well-understood genetic makeup, making it an ideal candidate for experimental research.
- **Social Behavior**: The interactions between fruit flies have a significant impact on their behavior, including mating, grooming , and movement. By studying these interactions, researchers can gain a deeper understanding of the movement. By studying these interactions, researchers benefits of cooperative behavior.
- **Computational Advance**: Recent breakthroughs in computational methods, such as Dynamic Bayesian Networks and Markov chains, have enabled researchers to effectively model and analyze behavioral data. These techniques can capture the random nature of interactions and predict future behaviors based on observed data.

B. Motivation

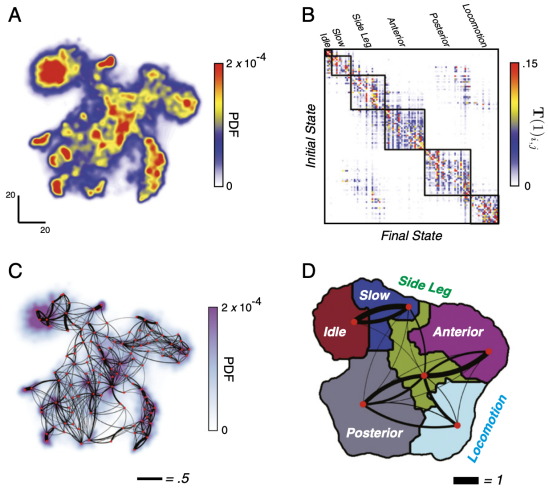
- **Understanding Interactions**: This project aims to us probabilistic methods to analyze the social interactions of fruit flies, helping to reveal patterns and predict behaviors that may not be immediately apparent through traditional observational methods.
- **Application of Theory**: By working with the real datasets, students can apply theoretical concepts from the probability course to practical problems, enhancing their understanding of probabilistic modeling in biological system.
- **Broader Implications**: The insights gained from this research could inform studies in other species, contributing to a deeper understanding of social dynamics and interactions across various biological contexts, including potential implications for human behavior.
- **Skill Development**: This project provides students with essential skills in mathematical modeling, data analysis, and computational techniques, bridging the gap between theoretical knowledge and practical application in computational biology and related fields

C. Applications

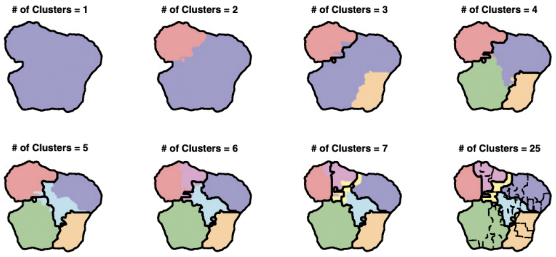
- **Behavioral Ecology**: Social interactions seen within groups of fruit flies provide further understanding of potential evolutionary benefits of social behaviors and can give a broader perspective when studying cooperation and competition.
- **Neuroscience Research**: The interactions modeled herein can serve as rich behavioral data sets that can lend insight into the neural components ofsocial behavior, increasingly relevant for neuropsychiatric disorders in more complex animals.
- **Machine Learning and AI**: The resulting probabilistic models may find use for a wide range of tasks in machine learning, such as in the fields of pattern recognition and predictive modeling, to name a few (e.g., social network analysis and behavior prediction).

  Public Health and Epidemiology: Results may have implications for public health by illuminating the role of social interactions in human disease spread and health behavior.
- **Interdisciplinary Research**: The project encourages research across biology, computer science, and social scientists addressing population dynamics and complex systems.
- **Genetic Studies**: The research will identify genetic influences on social behaviour,revealing the heritability and evolutionary significance of these traits.
- **Robotics**: Learning from the behavior of fruit flies, such as how they coordinate during mating, can inspire algorithms for swarm robotics that enable better coordination and cooperation among robotic units.
- **Education**: The results can be employed for educational purposes, e.g., to teach concepts in biology and computational science with an engaging real-world application.

II. T1 - Mathematical Modelling and Mathematical Analysis



(a) fig 1



(b) fig 2

1. Behavioral State Space (z)

Every point in the space corresponds to stereotypical behaviour like walking,grooming,etc and this is mapped into a 2D behavioral space using unsupervised learning. Defining a Probability Distribution Function over this space:

$$P(z) \tag{1}$$

In the resulting behavioral space, z, we estimate the probability distribution function P(z) and find that it contains a set of peaks corresponding to short segments of movement that are revisited multiple times by multiple individuals (Fig. 1A).

2. Transition Matrix

The matrix T (τ=1) describes the probability of transitions from one state to the next, the most elementary steps of behavior (Fig. 1B). By appropriately organizing the states in Fig. 1B, T(τ=1) takes on a nearly block-diagonal structure, which can be broken up into modular clusters using the information bottleneck formalism.

$$P(S(n+1) \mid S(n)) \tag{2}$$

where:

- The behavioral state at time n is $S(n)$
- The probability of transitioning from state $S(n)$ to $S(n+1)$ is $P(S(n+1) \mid S(n))$

3. Hierarchical Clustering of Behaviors

If the states are grouped into a hierarchy, we expect that increasing the number of clusters will largely subdivide existing clusters rather than mix behaviors from two different clusters (fig 1-D and fig 2).

$$S(n) \to Z \tag{3}$$

where:

- The fine-grained behavior at time n is $S(n)$
- A cluster or abstract behavior group is Z:

4. Information-Theoretic Optimization

The intention is to maintain the ability to predict future behavior while compressing behavioral description. This is done by maximizing the information about the future behaviour while holding the information we have about the past.

$$\max I(Z; S(n+\tau)) \quad \text{keeping} \quad I(Z; S(n)) = \text{const} \tag{4}$$

where:

- The information between cluster Z and future behavior is $I(Z; S(n+\tau))$
- The information between cluster Z and current behavior is $I(Z; S(n))$
- The time delay into the future is τ

5. Predictive Information

The quantification of how much current behavior tells us about the future is obtained by predictive information.

$$I_{\text{pred}} = I(S(n); S(n+\tau)) \tag{5}$$

6. Treeness Metric T

A "Treeness" metric is defined to quantify the hierarchical structure of behavioral clusters.

$$T = \frac{H_{\text{fwd}} - H_{\text{bwd}}}{H_{\text{fwd}}} \tag{6}$$

where:

- Entropy of paths traced forward is H_{fwd}
- Entropy of paths traced backward is H_{bwd}

Interpretation of T:

- $T = 1$: perfect hierarchy
- $T = 0$: no hierarchy

III. T2 - Code (with description of each line)

```
% Load datasets
courtshipData = readtable('courtship_complete.csv');
maleData = readtable('male_complete.csv');
femaleData = readtable('female_complete.csv');

% Function to compute correlation
compute_corr = @(X, Y) sum((X - mean(X)) .* (Y - mean(Y))) / ...
    sqrt(sum((X - mean(X)).^2) * sum((Y - mean(Y)).^2));

% Compute behavioral correlations
corr_courtship = compute_corr(courtshipData.Fly1_behavior,
   courtshipData.Fly2_behavior);
corr_male = compute_corr(maleData.Fly1_behavior, maleData.
   Fly2_behavior);
corr_female = compute_corr(femaleData.Fly1_behavior, femaleData.
   Fly2_behavior);

% Display results
fprintf('Courtship Pairs - Behavioral Correlation: %.5f\n',
   corr_courtship);
fprintf('Male-Male Pairs - Behavioral Correlation: %.5f\n',
   corr_male);
fprintf('Female-Female Pairs - Behavioral Correlation: %.5f\n',
   corr_female);


%% FUNCTION TO PLOT BEHAVIOR TRANSITION HEATMAP
function plotTransitionHeatmap(fly1_behavior, fly2_behavior,
   titleText)
    % Get unique states
    uniqueStates = unique([fly1_behavior; fly2_behavior]);
    numStates = length(uniqueStates);

    % Create an empty transition matrix
    transitionMatrix = zeros(numStates);

    % Populate transition matrix
    for i = 1:length(fly1_behavior)-1
        row = find(uniqueStates == fly1_behavior(i));
        col = find(uniqueStates == fly2_behavior(i));
        transitionMatrix(row, col) = transitionMatrix(row, col) + 1;
    end

    % Normalize
    transitionMatrix = transitionMatrix ./ max(1, sum(
       transitionMatrix, 2));
```

```
39
40      % Plot heatmap
41      figure;
42      heatmap(uniqueStates, uniqueStates, transitionMatrix);
43      title([titleText, ' - Behavior Heatmap']);
44      xlabel('Fly 1 Behavior');
45      ylabel('Fly 2 Behavior');
46  end
47
48  % Plot transition heatmaps
49  plotTransitionHeatmap(courtshipData.Fly1_behavior, courtshipData.
        Fly2_behavior, 'Courtship Pairs');
50  plotTransitionHeatmap(maleData.Fly1_behavior, maleData.Fly2_behavior
        , 'Male-Male Pairs');
51  plotTransitionHeatmap(femaleData.Fly1_behavior, femaleData.
        Fly2_behavior, 'Female-Female Pairs');
```

Listing 1: Behavioral Correlation and Visualization in MATLAB

A. Code Explaination:

1. **These lines of the code load the data for the 3 condition:**

   - The Courtship pairs (i.e., a male fly and a female fly)
   - The Male pairs (2 male flys)
   - The Female pairs (2 female flys)

```
1  courtshipData = readtable('courtship_complete.csv');
2  maleData = readtable('male_complete.csv');
3  femaleData = readtable('female_complete.csv');
```

Listing 2: Loading the datasets

2. **Function to compute correlation:**

   - This one line is a function of calculating Pearson's correlation coefficient in MATLAB
   - It is to measure how synchronized behavior does both the fly show
   - It is helpful in studying how the behavior of one fly variates or affects the behavior of the other fly

```
1  compute_corr = @(X, Y) sum((X - mean(X)) .* (Y - mean(Y))) /
       ...
2      sqrt(sum((X - mean(X)).^2) * sum((Y - mean(Y)).^2));
```

Listing 3: Loading the datasets

3. **Understanding how behavior changes among the pairs:**

```
corr_courtship = compute_corr(courtshipData.Fly1_behavior,
   courtshipData.Fly2_behavior);
corr_male = compute_corr(maleData.Fly1_behavior, maleData.
   Fly2_behavior);
corr_female = compute_corr(femaleData.Fly1_behavior, femaleData
   .Fly2_behavior);
```

Listing 4: Loading the datasets

4. **Display results to compare the social behavior:**

```
fprintf('Courtship Pairs - Behavioral Correlation: %.5f\n',
   corr_courtship);
fprintf('Male-Male Pairs - Behavioral Correlation: %.5f\n',
   corr_male);
fprintf('Female-Female Pairs - Behavioral Correlation: %.5f\n',
    corr_female);
```

Listing 5: Loading the datasets

5. **Function to plot heatmap:**

```
function plotTransitionHeatmap(fly1_behavior, fly2_behavior,
   titleText)
```

Listing 6: Loading the datasets

6. **Inside the function:**

- It finds all the possible unique behaviors across both the flies
- Defines axis for the heatmap
- It stores the transition counts by initializing an empty square matrix

```
uniqueStates = unique([fly1_behavior; fly2_behavior]);
numStates = length(uniqueStates);
transitionMatrix = zeros(numStates);
```

Listing 7: Loading the datasets

- The matrix is filled by counting that how many of the fly 1 behavior is paired with that of fly 2 behavior
- The synchronous behavior transitions are analogous to a co-occurrence matrix
- It normalize each row, so that they correspond to relative frequencies

```
for i = 1:length(fly1_behavior)-1
    row = find(uniqueStates == fly1_behavior(i));
    col = find(uniqueStates == fly2_behavior(i));
    transitionMatrix(row, col) = transitionMatrix(row, col)
       + 1;
end
transitionMatrix = transitionMatrix ./ max(1, sum(
   transitionMatrix, 2));
```

Listing 8: Loading the datasets

7. **Plots heatmap:**

- The rows show the Fly 1 behavior
- The column shows the Fly 2 behavior
- The frequency of occurrence show the intensity of color
- This then finally generates the heatmaps

```
figure;
heatmap(uniqueStates, uniqueStates, transitionMatrix);
title([titleText, ' - Behavior Heatmap']);
xlabel('Fly 1 Behavior');
ylabel('Fly 2 Behavior');

plotTransitionHeatmap(courtshipData.Fly1_behavior,
   courtshipData.Fly2_behavior, 'Courtship Pairs');
plotTransitionHeatmap(maleData.Fly1_behavior, maleData.
   Fly2_behavior, 'Male-Male Pairs');
plotTransitionHeatmap(femaleData.Fly1_behavior, femaleData.
   Fly2_behavior, 'Female-Female Pairs');
```

Listing 9: Loading the datasets

IV. T4 - Algorithm (Deterministic/Baseline and Randomized)

Algorithm 1 PC algorithm for Bayesian structure learning

Require: Variables $X = \{X_1, X_2, \ldots, X_n\}$
Ensure: A DAG C representing the conditional independence structure of X

Initialization: Form a complete undirected graph C where each pair of variables in X is connected by an edge.
Initialization: $n = 0$
while there exists an edge between X_i and X_j in C such that $|Adj(C, X_i) \setminus \{X_j\}| \geq n$ **do**
    for all pairs of variables (X_i, X_j) in C **do**
        for all subsets $S \subseteq Adj(C, X_i) \setminus \{X_j\}$ with $|S| = n$ **do**
            if X_i and X_j are conditionally independent given S **then**
                Remove edge $X_i - X_j$ from C
            end if
        end for
    end for
    $n = n + 1$
end while
for all pairs of non-adjacent nodes (X_i, X_k) both adjacent to X_j **do**
    if separating set S of (X_i, X_k) does not contain X_j **then**
        Orient $X_i - X_j - X_k$ as $X_i \to X_j \leftarrow X_k$
    end if
end for
for all $X_i \to X_j \to X_k$ **do**
    if X_i and X_k are adjacent **then**
        Orient $X_i - X_k$ as $X_i \to X_k$
    end if
end for
return The new DAG C

Overall Time Complexity:

$$\mathcal{O}(n^2 \cdot 2^d)$$

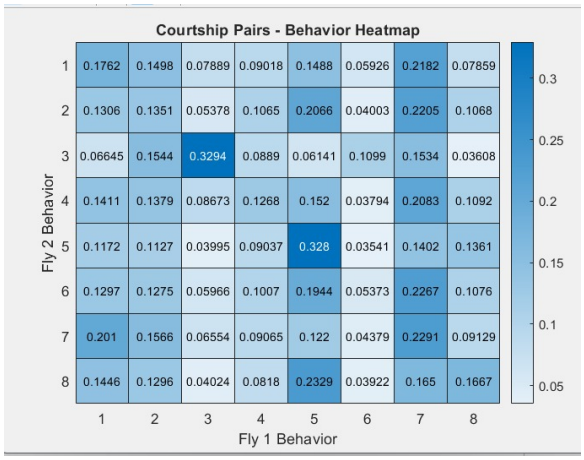
Here, n is the number of variables and d is the maximum node degree in the graph during execution. The exponential term arises due to evaluating all subsets of neighbors of size up to d for each pair (i, j).

Algorithm 2 Randomized PC Algorithm with Bootstrap for Bayesian Network Structure Learning

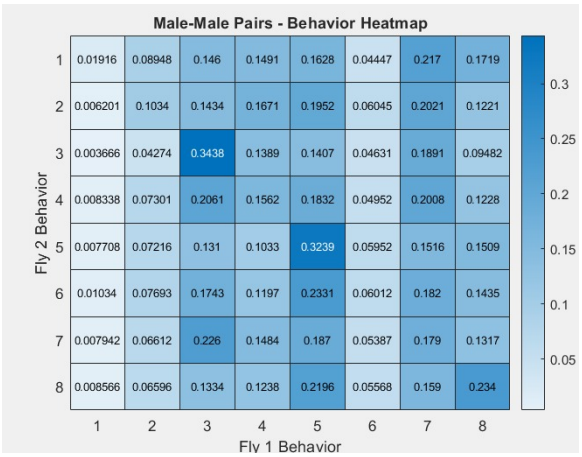
Require: Dataset D, Significance range $\alpha_{\text{range}} = (\alpha_{\min}, \alpha_{\max})$, Bootstrap flag `bootstrap`, Number of bootstraps n_{boot}
Ensure: Approximate skeleton C, Separation Sets S, Directed Acyclic Graph G_{DAG}

Step 1: Randomize Variable Order
Randomly interchange the variable locations in D
Step 2: Randomize Significance Level
Sample $\alpha \sim U(\alpha_{\min}, \alpha_{\max})$
Step 3: Skeleton Learning
if `bootstrap` is `True` **then**
    Initialize edge frequency dictionary `edge_counts`
    for $i = 1$ to n_{boot} **do**
        Generate bootstrap sample D_i from D
        Learn skeleton G_i on D_i with significance level α
        for each edge $(u, v) \in G_i$ **do**
            Increment `edge_counts`$[(u, v)]$ by 1
        end for
    end for
    Construct undirected graph G by adding edges with frequency $> 50\%$
else
    Learn undirected skeleton G on D with significance level α
end if
Step 4: Orient Edges to Build DAG
Initialize empty directed graph G_{DAG}
for each undirected edge $(u, v) \in G$ **do**
    Try orienting edge as $u \to v$ in G_{DAG}
    if cycle is formed **then**
        Try orienting edge as $v \to u$
        if still cyclic **then**
            Remove edge (u, v)
        end if
    end if
end for
Step 5: Identify Colliders (V-Structures)
for each node z in variables **do**
    for each pair of neighbors (x, y) of z **do**
        if x and y are independent and edges $x \to z$ and $y \to z$ exist **then**
            Keep collider structure: $x \to z \leftarrow y$
        end if
    end for
end for
return G_{DAG}

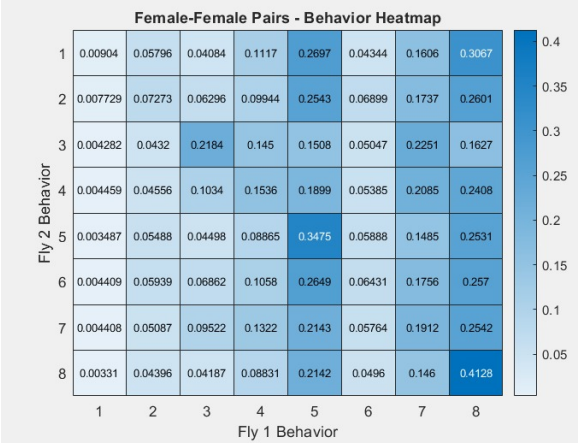
V. T3 - Results and Inferences (Domain + CS perspective)



(a) fig 1



(b) fig 2



(c) fig 3

Figure 1: **Correlation between behaviors performed by individuals within a pair**

A. Domian Perspective (Biological)

1. **Courtship Pairs (Fig. 1) - Strong Behavioral Synchronization**

   - We can see high values along the diagonal. This indicates strong self-synchrony.
   - This is seen in courtship pairs because of social signaling and reciprocation.
   - Hence, we can observe that social interactions between different sexes influence coordination. [**2**]

2. **Male - Male Pairs (Fig. 2) - Moderate Synchronization**

   - Here, the diagonal is less prominent. Some behaviors still have some visible peaks, but the off-diagonal values are not very pronounced.

- This indicates that even in pairs of the same sex, there is social awareness. However, synchronization is less than in courtship pairs.
- There is a possibility that they show competitive and avoidance-driven synchronization.

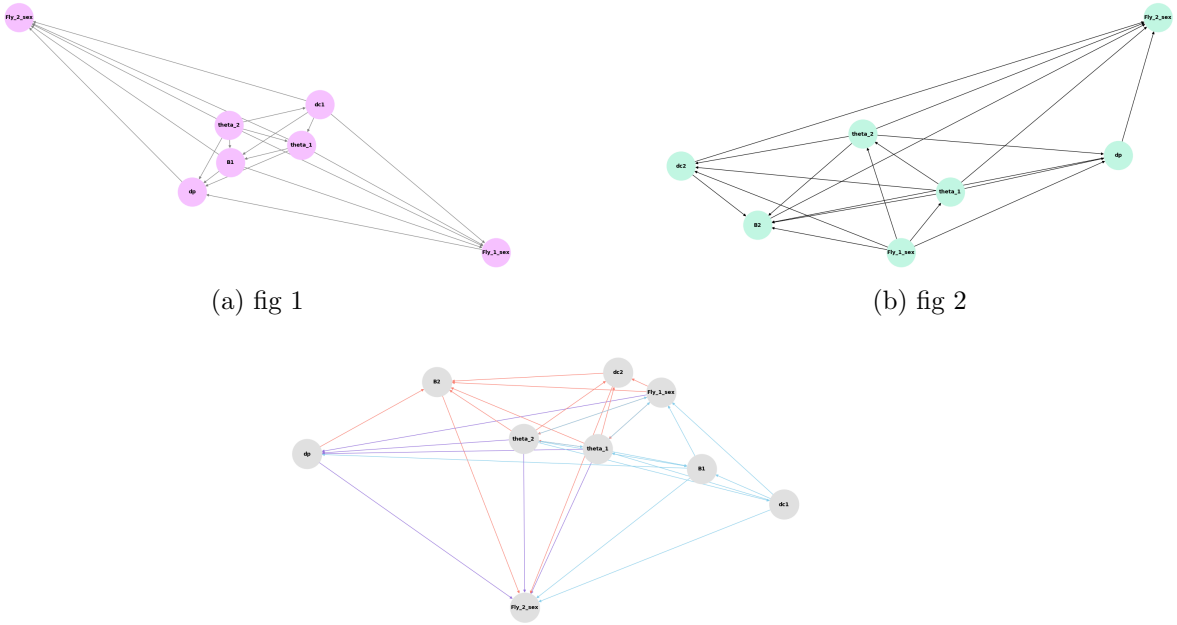
3. **Female - Female Pairs (Fig. 3) - Unexpected Synchronization**
   - We can see a strong diagonal dominance in this plot, which was unexpected.
   - We can see that female-female pairs may involve socially positive interactions and environmental regulation.
   - This may indicate that when one female is performing a certain behavior, there is a strong possibility that, at the same time, another female performs the same behavior.

B. CS Perspective (Data Science)

1. **Joint Probability Matrix - Heatmaps**
   - Here, we have used the joint probability distribution table to know the likelihood of variables (Fly 1 and Fly 2) happening together.
   - In heatmaps, we can see that the diagonal elements show self-synchrony. On the other hand, off-diagonal elements show compensatory behavior.
   - Such joint distribution tables can be used to learn the Bayesian Networks.

2. **Bayesian Networks**
   - This network is a probabilistic model (graphical) that represents a set of variables and their dependencies with the help of Directed Acyclic Graphs (DAG).
   - Dynamic Bayesian Networks are used for time-dependent traditions in behavior. It is also used to predict the next actions.

3. **Computational Comparisons**
   - The plots above (Figure 1) used more of a correlation approach rather than a Bayesian model.
   - These plots are static co-occurrence matrices that can be further enhanced by applying Dynamic Bayesian Networks.



(a) fig 1

(b) fig 2

(c) fig 3

Figure 2: **The trio of probing, mirroring, and merging methods is exhibited on our model.**

C. Domian Perspective (Biological)

- Here, the graphs show the casual relationships with the help of Dynamic Bayesian Networks.
- The graph with combined network higher interconnectivity. Here, the sex of the flies are the target nodes, which are influenced by other nodes. Thus, the sex of the fly is affected by and affects the motion parameters.
- The complexity of the dependency structures is comparatively less in mirror interactions.
- In the probe interactions, we can see that flies continuously respond to other flies' sex and state of motion.

D. CS Perspective (Data Science)

- All these graphs show directed dependencies. Such graphs are ideal for temporal or causal interpretation.
- The sparse vs. dense graph differences are used to check how model complexity adapts to the data structure.
  - Probe (fig. 1) - High interactions
  - Mirror (fig. 2) - Low interactions
  - (fig. 3) - Overlapping interactions

VI. T5 - Derivation of Bounds and Results (new inferences)

- The PC algorithm uses tests like mutual information to find relationships between variables.
- If only a small number of fruit fly behaviors are observed, the causal graph it creates might not be accurate. This is Because the accuracy of the PC algorithm depends on how many conditional independence tests it can perform.
- The error in estimating mutual information is bounded by:

$$|I(B_i; B_j|S) - \hat{I}(B_i; B_j|S)| \leq O\left(\frac{1}{\sqrt{N}}\right)$$

- The PC algorithm performs better when the number of samples is large, reducing the false discovery rate of inferred causal relationships.

Case	**Time Complexity**
Best Case (Sparse Graph)	$O(n^2)$
Average Case (Moderate Density Graph)	$O(n^k)$, where k is the maximum degree
Worst Case (Dense Graph)	$O(n \cdot 2^n)$

Table 1: Time Complexity of the PC Algorithm

- The PC algorithm creates a Directed Acyclic Graph (DAG) by checking if variables are conditionally independent, assuming that all important factors are included.
- PC-Stable improves how consistently the connections between variables are drawn, so the results stay the same if the process is repeated. It does this without making the algorithm more complicated.
- The Fast Causal Inference (FCI) algorithm builds on PC and takes into account hidden variables and selection bias. instead of a DAG, it produces a Partial Ancestral Graph(PAG). Although FCI works better in real-world situations, it requires more computing power.

Property	**PC Algorithm**	**PC-Stable**	**FCI**
Assumes Causal Sufficiency	Yes	Yes	No
Handles Latent Variables	No	No	Yes
Handles Selection Bias	No	No	Yes
Computational Complexity (WorstCase)	$O(n \cdot 2^n)$	$O(n \cdot 2^n)$	$O(n^2 \cdot 2^n)$
Produces DAG	Yes	Yes	No (Produces PAG)
Edge Orientation Accuracy	Can be unstable	More stable	More reliable
Suitable for Dense Graphs	No	No	Partially
Parallel Execution Support	Limited	Yes	No

Table 2: Comparison of PC, PC-Stable, and FCI Algorithms

References

[1] Kalisch, M., Buhlmann, P. (2007). *JMLR. Estimating High-Dimensional Directed Acyclic Graphs with the PC-Algorithm.* https://www.jmlr.org/papers/volume8/kalisch07a/kalisch07a.pdf

[2] Bornstein, M. H., Esposito, G. (2023, July 31). *Coregulation: A multilevel approach via biology and behavior. MDPI.* https://www.mdpi.com/2227-9067/10/8/1323

[3] **PC (constraint-based estimator)¶. PC (Constraint-Based Estimator) - pgmpy 1.0.0 documentation. (n.d.).** https://pgmpy.org/structure_estimator/pc.html

[4] Berman, G. J., Bialek, W., Shaevitz, J. W. (2016, October 4). *PNAS.* https://www.pnas.org/doi/epdf/10.1073/pnas.1607601113

[5] Tolwinski, N. S. (2017, September 20). *Introduction: Drosophila-A model system for developmental biology. MDPI.* https://www.mdpi.com/2221-3759/5/3/9

[6] Kalaria, K., Mayekar, H., Patel, D., Rajpurohit, S. (2023, January 1). *Dynamic Bayesian modeling of the social behavior of drosophila melanogaster. bioRxiv.* https://www.biorxiv.org/content/10.1101/2023.03.02.530782v1.full

[7] *Klibaite, U., Shaevitz, J. W. (n.d.). Paired fruit flies synchronize behavior: Uncovering social interactions in drosophila melanogaster. PLOS Computational Biology.* https://journals.plos.org/ploscompbiol/article?id=10.1371%2Fjournal.pcbi.1008230

[8] Chen, X., Huang, Q., Zhou, Y. (2024a, November 18). *Transmission line outage probability prediction under extreme events using Peter-Clark Bayesian Structural Learning. arXiv.org.* https://arxiv.org/abs/2411.11980