__/__/____

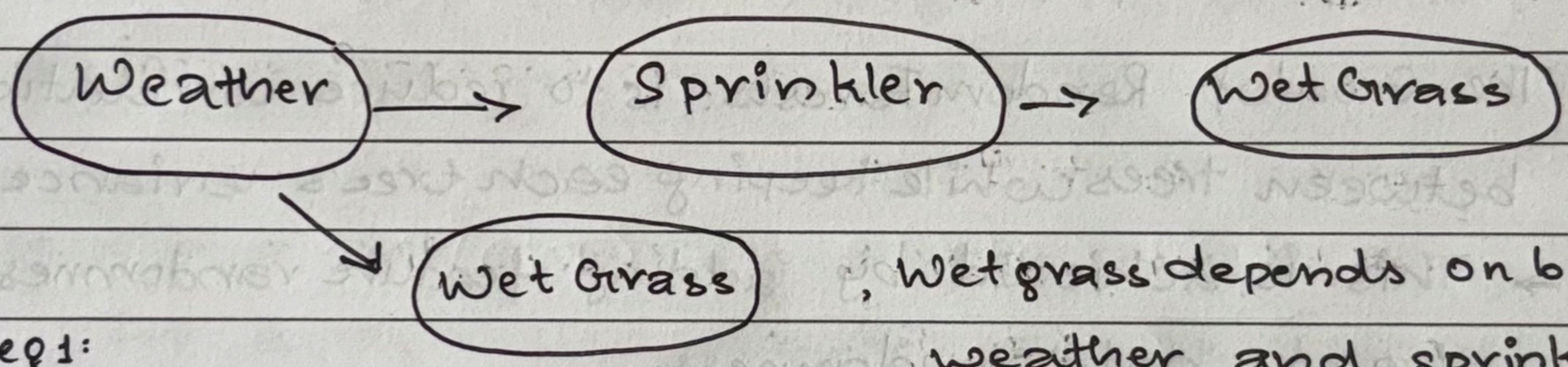
(31) Bayesian Networks

A Bayesian Network is a graphical Model that represents the joint probability distribution of a set of random variables using a Directed Acyclic Graph.

To infer probabilities of certain events given evidance.

- Each node: a random variable
- Each edge: a direct probabilistic dependency

Each node has a Conditional Probability Table (CPT) that specifies
→ how that variable depends on its parent nodes.



eg1: , Wet grass depends on both weather and sprinkler.

- Joint Probability Distribution

Joint Probability of all variables = product of local conditional probabilities.

If we've variables $x_1, x_2 \dots x_n$
and each variable depends only on its parents in the graph:

$$P(x_1, x_2 \dots x_n) = \prod_{i=1}^{n} P(x_i \mid \text{Parents}(x_i))$$

- Chain rule for Bayesian Networks

Joint Probability for eg1:

$$P(\text{Weather}, \text{Sprinkler}, \text{Grass}) = P(W) \times P(S|W) \times P(G|S, W)$$

* Factorization: Product over conditional probability
Typical Use: Casual Models

__/__/____

- Conditional Independence

  Bayesian Networks capture conditional independence automatically through structure.

  (Sprinkler ⫫ Weather | WetGrass)

  If there is no direct or indirect path between two nodes they are conditionally independent.

- d- seperation : helps determine if two nodes are conditionally independent or not.

1. Chain: $A \rightarrow B \rightarrow C$

   A & C are dependent, but if B is known, they're independent.

2. Fork: $A \leftarrow B \rightarrow C$

   A & C are dependent, independent if B is known.

3. Collider: $A \rightarrow B \leftarrow C$

   A & C are independent, but become dependent if B (or descendant of B) is known.

Advs:
1. Probabilistic Reasoning
2. Handles missing data
3. Visual and interpretable

Disadvs:
1. Structure learning is hard
2. Require lots of data
3. Inference can be slow for large networks.

__/__/____

(32) Naive Bayes

It is a probabilistic classification model based on Bayes Theorem with a strong independent assumption:

All features (predictors) are conditionally independent given the class label.

→ It is "naive" because this assumption rarely holds perfectly in real data - but it works surprisingly well!

- Bayes Theoram

C = class Label , $X = (x_1, x_2, \dots x_n)$ = feature vector

$$P(C|x) = \frac{P(x|C)\,P(C)}{P(x)}$$

Posterior = likelihood × prior / evidence

- Naive Independence Assumption

$$P(x|C) = P(x_1, x_2 \dots, x_n \mid C) = \prod_{i=1}^{n} P(x_i \mid C)$$

This simplifies computation drasticelly

$$P(C|x) \propto P(C) \prod_{i=1}^{n} P(x_i|C)$$

The denominator $P(X)$ is constant for all classes, so we only compute the numerators.

* Posterior ∝ Prior * Likelihood
* Naive Bayes often beats complex models on small datasets.

__/__/____

- Classification Rule

Predict class $\hat{c}$ as: $\hat{C} = \arg\max_c P(c) \prod_{i=1} P(x_i|C)$

taking log to avoid underflow.

$\tilde{C} = \arg\max \left[\log P(c) + \sum_i \log P(x_i|c) \right]$

→ Types of Naive Bayes Classifiers

1. Bernoulli Naive Bayes: Used for Binary features

$P(x_i|c) = p_i^{x_i} (1-p_i)^{1-x_i}$

2. Multinomial Naive Bayes: Used for count features

$P(x|c) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i p_i^{x_i|c}$

3. Gaussian Naive Bayes: When features are continuous and modelled with Gaussian Distribution.

$$P(x_i|c) = \frac{1}{\sqrt{2\pi\sigma^2_{c,i}}} \exp\left(-\frac{(x_i - \mu_{c,i})^2}{2\sigma^2_{c,i}}\right)$$

- Smoothing (Laplace Correction)

When a feature never appears in training data for a class.

$P(x_i|C) = 0$ → entire product becomes 0.

To fix this: $P(x_i|c) = \frac{\text{count}(x_i, c) + 1}{\text{count}(c) + k = \text{no of possible values.}}$

This is Laplace (add-one) smoothing.