Advance Machine Learning

01CO1301

4 Credits

Marwadi University

Department of Computer Engineering

Ravikumar R Natarajan

Course Outcomes

- At the end of the course, students will be able to:
- To understand key concepts, tools and approaches for pattern recognition on complex data sets.
- To learn Kernel methods for handling high dimensional and non-linear patterns.
- To implement state-of-the-art algorithms such as Support Vector Machines and Bayesian networks.
- To Solve real-world machine learning tasks: from data to inference.
- To apply theoretical concepts and the motivations behind different learning frameworks.

Marwadi University

Department of Computer Engineering

Bayesian Methods

Unit #3

Content

- Bayesian Methods for using Prior Knowledge and Data
- Bayesian Inference
- Bayesian Belief Networks and Graphical Models
- Probabilistic Latent Semantic Analysis
- The Expectation-Maximisation (EM) Algorithm
- Gaussian Processes

Bayes Theorem vs Bayesian Method

- Bayes theorem helps to determine the probability of an event with random knowledge. It is used to calculate the probability of occurring one event while other one already occurred.
- It is a best method to relate the condition probability and marginal probability.

- Bayesian method is used to calculate conditional probability in Machine Learning application that includes classification tasks.

Bayesian Belief Network

Most of you may already be familiar with the Naive Bayes algorithm, a fast and simple modeling technique used in classification problems. While it is used widely due to its speed and relatively good performance, **Naive Bayes is built on the assumption that all variables (model features) are independent, which in reality is often not true.**

In some cases, you may want to build a model where you can specify which variables are dependent, independent, or conditionally independent (this is explained in the next section). You may also want to track real-time how event probabilities change as new evidence is introduced to the model.

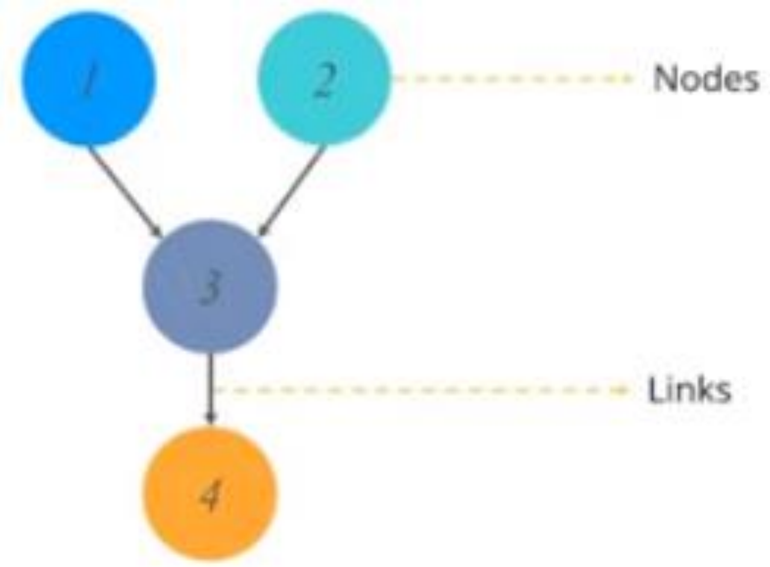
This is where the Bayesian Belief Networks come in handy as they allow you to construct a model with nodes and directed edges by clearly outlining the relationships between variables.

What is Bayesian Network?

A Bayesian Network (BN) falls under the category of Probabilistic Graphical Modelling (PGM) technique that is used to compute uncertainties by using the concept of probability.

DAG (Directed Acyclic Graph)

- It is used to represent BN.
- DAG contains set of nodes and links, where the links denote the relationship between nodes.



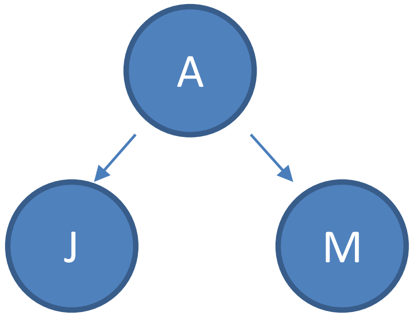
Conditional Probability and Joint Probability

Conditional Probability – measure of the probability of event B where event A has already occurred.

P(A n B) = p(A/B).P(B)
P(B n A) = p(B/A).p(A)
P(B/A) = P(A/B).P(B)/P(A)

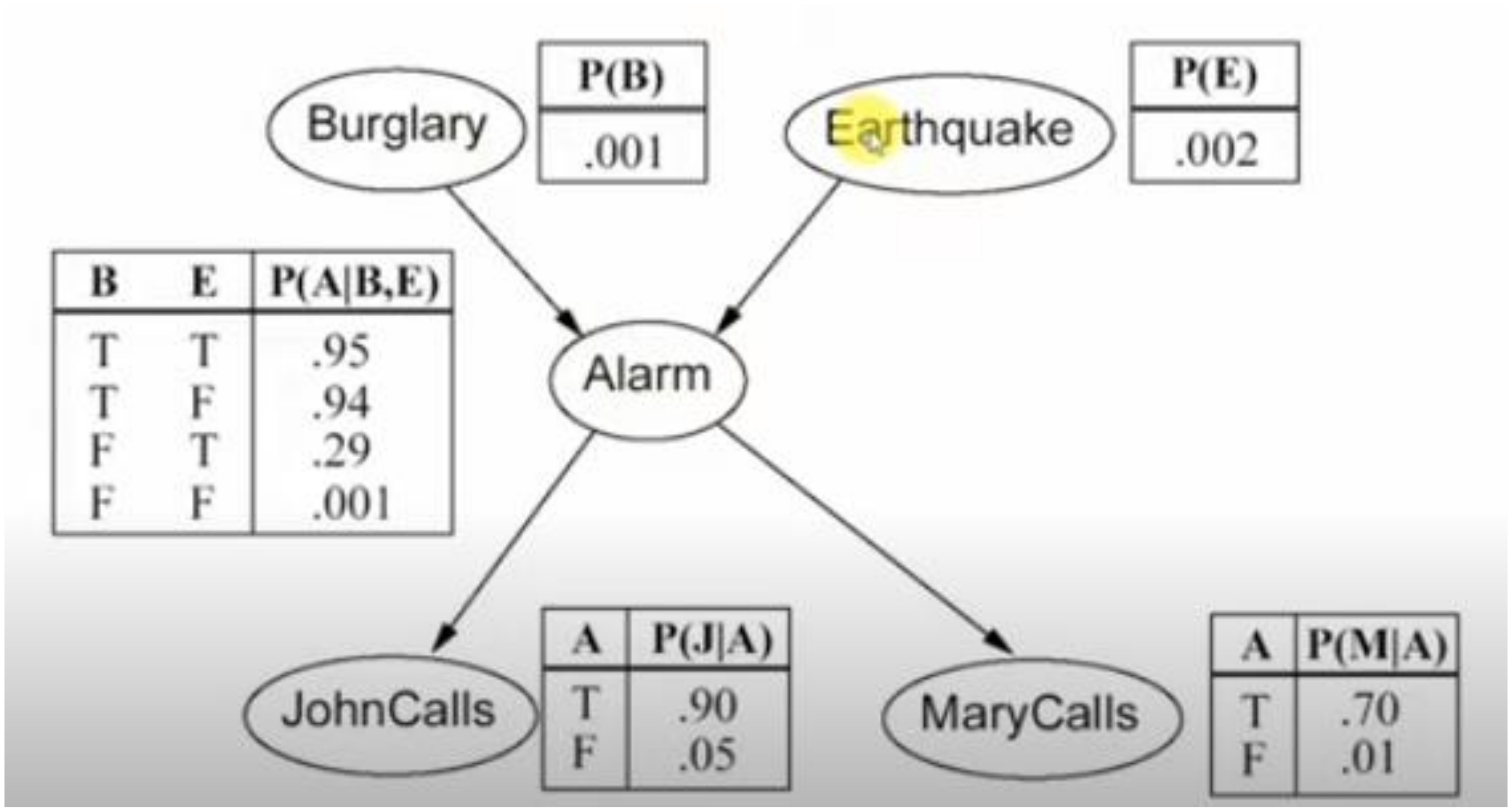
Joint Probability – measure of two events happening at same time.

$$P(x1, x2, \ldots\ldots\ xn) = \prod_{i=1}^{n} (xi/parent(x))$$



P(A,J,M) = P(A)*P(J/A)*P(M/A)

Bayesian Belief Network



P(B)
.001

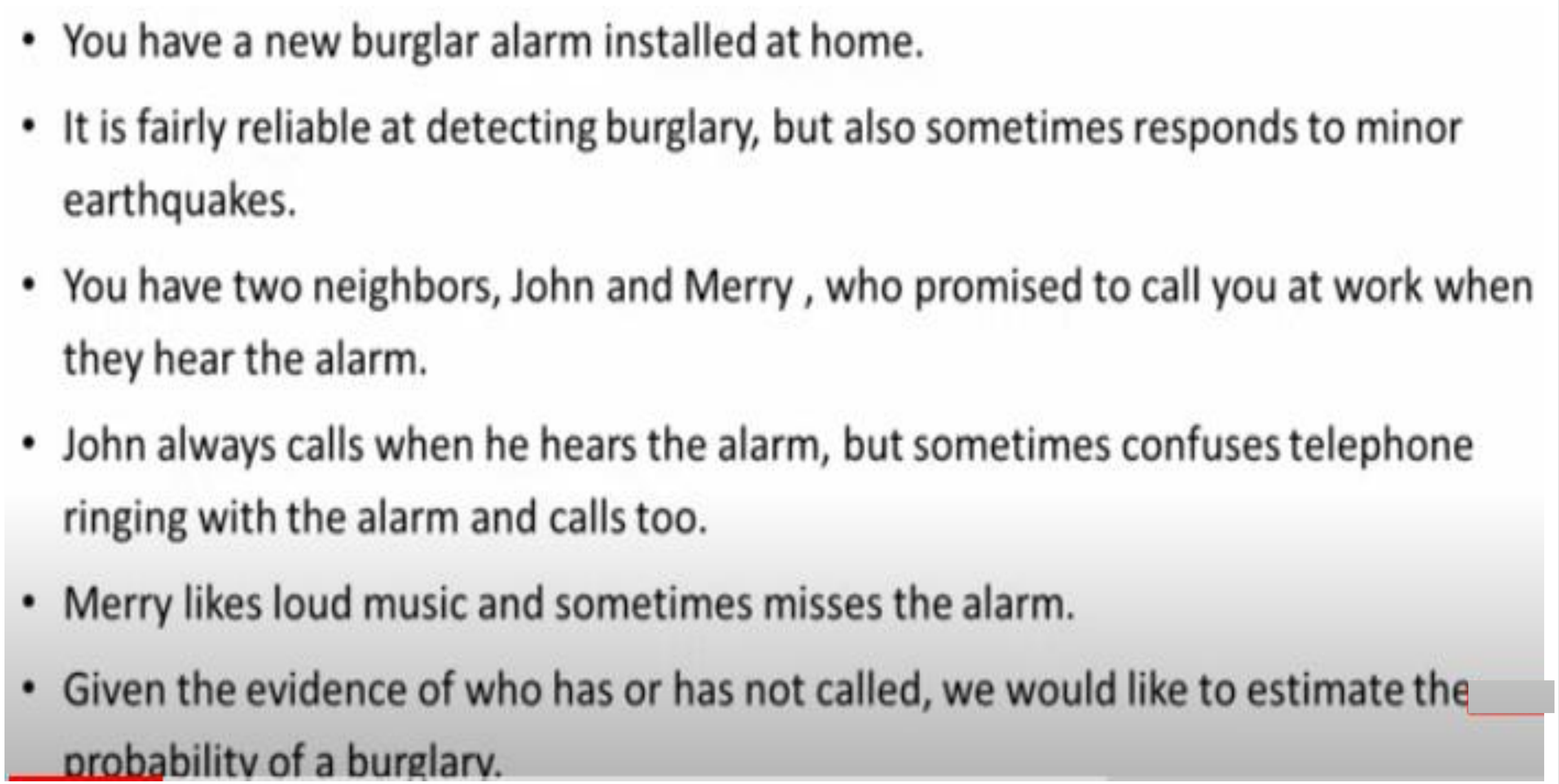
P(E)
.002

B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J|A)
T	.90
F	.05

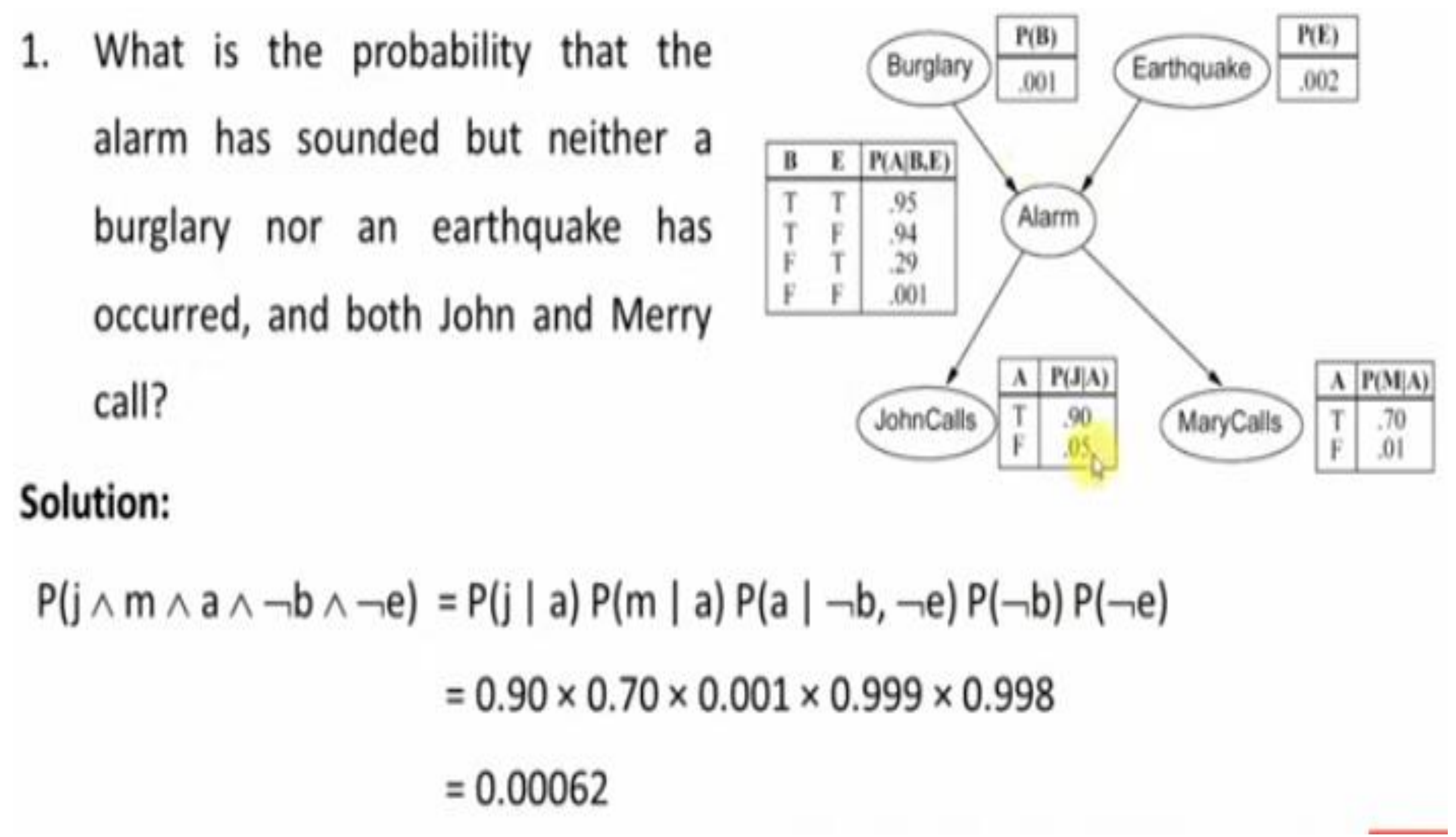
A	P(M|A)
T	.70
F	.01

Bayesian Belief Network



- You have a new burglar alarm installed at home.
- It is fairly reliable at detecting burglary, but also sometimes responds to minor earthquakes.
- You have two neighbors, John and Merry , who promised to call you at work when they hear the alarm.
- John always calls when he hears the alarm, but sometimes confuses telephone ringing with the alarm and calls too.
- Merry likes loud music and sometimes misses the alarm.
- Given the evidence of who has or has not called, we would like to estimate the probability of a burglary.

Bayesian Belief Network



1. What is the probability that the alarm has sounded but neither a burglary nor an earthquake has occurred, and both John and Merry call?

P(B)
.001

P(E)
.002

B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J|A)
T	.90
F	.05

A	P(M|A)
T	.70
F	.01

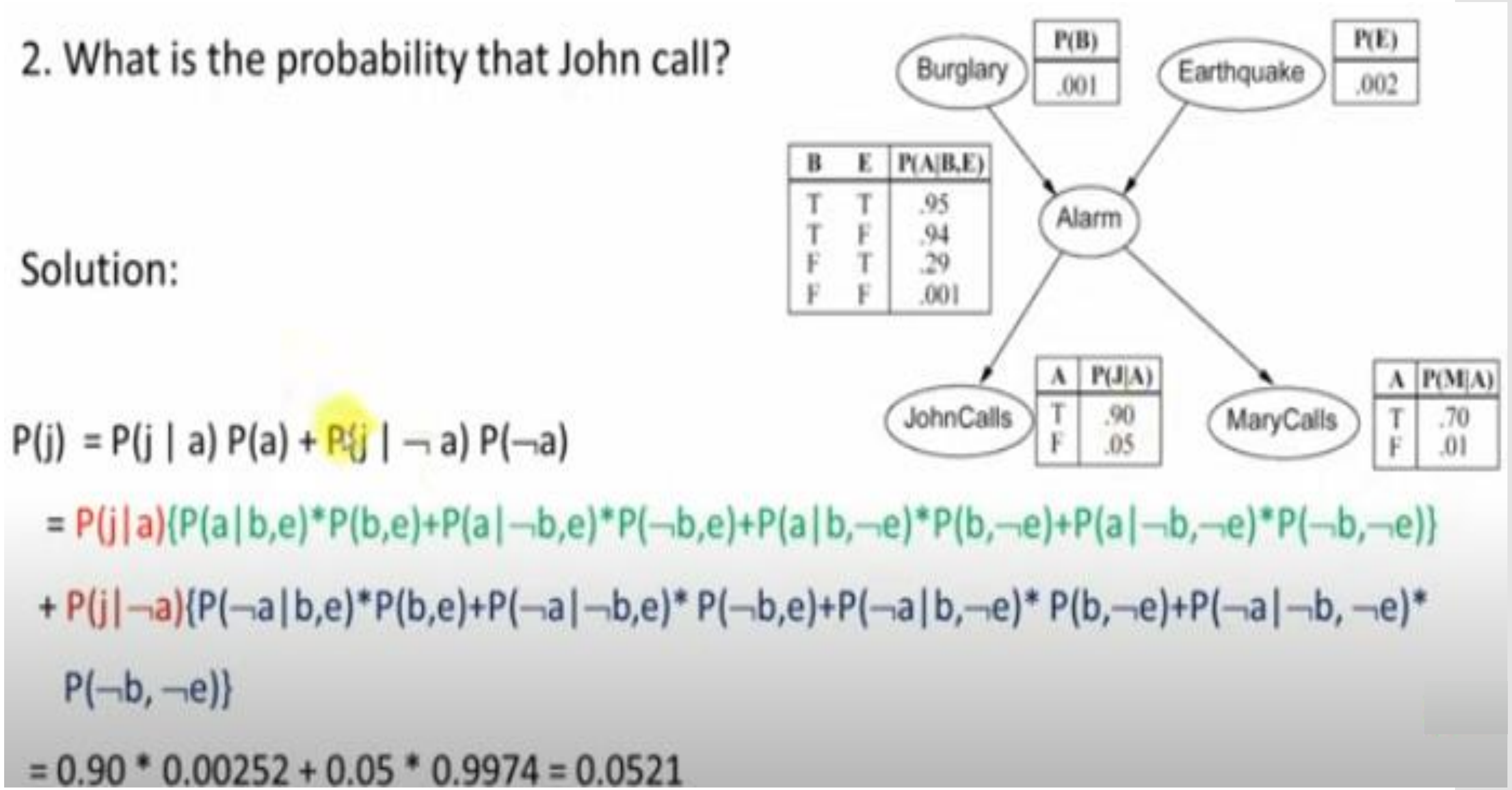
Solution:

$$P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) = P(j \mid a)\, P(m \mid a)\, P(a \mid \neg b, \neg e)\, P(\neg b)\, P(\neg e)$$

$$= 0.90 \times 0.70 \times 0.001 \times 0.999 \times 0.998$$

$$= 0.00062$$

Bayesian Belief Network



2. What is the probability that John call?

P(B)
.001

P(E)
.002

B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J|A)
T	.90
F	.05

A	P(M|A)
T	.70
F	.01

Solution:

$$P(j) = P(j \mid a)\,P(a) + P(j \mid \neg a)\,P(\neg a)$$

$$= P(j|a)\{P(a|b,e)*P(b,e)+P(a|\neg b,e)*P(\neg b,e)+P(a|b,\neg e)*P(b,\neg e)+P(a|\neg b,\neg e)*P(\neg b,\neg e)\}$$

$$+ P(j|\neg a)\{P(\neg a|b,e)*P(b,e)+P(\neg a|\neg b,e)*P(\neg b,e)+P(\neg a|b,\neg e)*P(b,\neg e)+P(\neg a|\neg b,\neg e)*P(\neg b,\neg e)\}$$

$$= 0.90 * 0.00252 + 0.05 * 0.9974 = 0.0521$$

Bayesian Network Applications

- Disease Diagnosis
- Optimized Web Search
- Spam Filtering
- Gene Regulatory Networks
- Bio-monitoring

Bayesian Belief Network Demo

- https://colab.research.google.com/drive/1FOS6c516iB9O35mPV9jMhzGgXG9q4U9t?usp=sharing

Probabilistic Latent Semantic Analysis PLSA

Suppose you have hundreds of articles/sentences. You want to know what topics each of those articles/sentences talk about. An article describing allegations made on a pharmaceutical company may talk about topics like Government, Medicine or Business. Our goal is to assign these topics to documents.

One of the methods to perform this task is Probabilistic Latent Semantic Analysis (PLSA).

PLSA or Probabilistic Latent Semantic Analysis is a technique used to model information under a probabilistic framework. Latent because the topics are treated as latent or hidden variables.

Variables involved in PLSA

- **Documents**
  - Representation: D={d1,d2,d3,…dN}
- **Words**
  - Representation: W={w1,w2,…wM}
- **Topic**
  - Representation: Z={z1,z2,…zk}

What is PLSA?

- Each document consists of a mixture of topics, and
- Each topic consists of a collection of words.

PLSA uses a probabilistic method instead of Singular Value Decomposition, which we used in LSA to tackle the problem.

The main goal is to find a probabilistic model with latent or **hidden topics** that can generate the data which we observe in our document-term matrix.

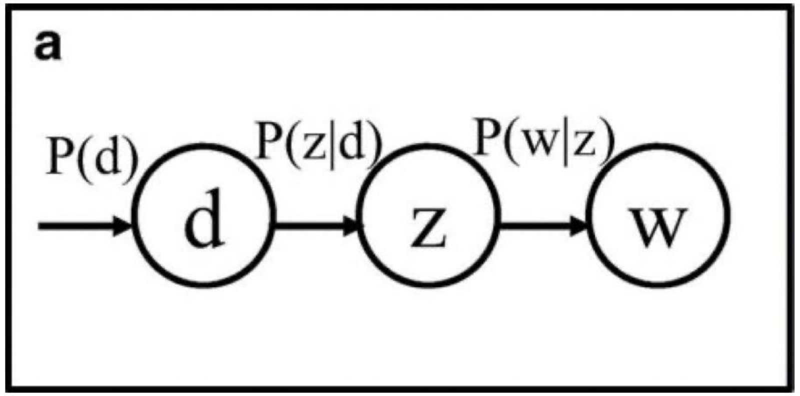
PLSA

- PLSA can be understood in two different ways.
    - **Latent Variable Model**
    - **Matrix Factorization**

Latent Variable Model for PLSA

Parameterization 1

In this parameterization, we sample a document first then based on the document we sample a topic, and based on the topic we sample a word, which means d and w are conditionally independent given a hidden topic ‘z’.



PLSA

Assumption-1(Bag of Words)

As we discussed while learning the text vectorization techniques that the word ordering in the vocabulary doesn’t matter. In simple words, the joint variable (d,w) is sampled independently.

$$P(\mathcal{D}, \mathcal{W}) = \prod_{(d,w)} P(d, w).$$

Assumption-2(Conditional Independence)

It is one of the key assumptions that we make while formulating the theory is that the words and the documents are conditionally independent. Focus on the word conditionally. This implies

P(w,d|z) = P(w|z)*P(d|z)

PLSA

Using conditional independence,

$$P(w|d) = \sum_{z \in Z} P(w|z)P(z|d)$$

Using Bayes Rule,

$$P(w,d) = \sum_{z \in Z} P(z)P(d|z)P(w|z).$$

PLSA

Matrix Factorization Model

An alternative way to represent PLSA is Matrix Factorization Model.

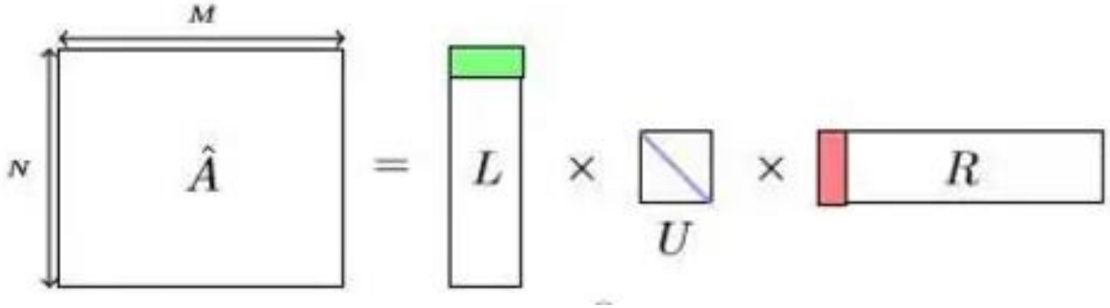
Consider a document-word matrix of dimensions N*M, where N is the number of documents and M is the size of the vocabulary. The elements of the matrix are counts of the occurences of a word in a document. If a word wi occurs once in the document dj, then element (j,i) = 1.

If you think of this matrix, most of the elements are 0. Say we have a document of 10 words and a vocabulary of 1000 words. Naturally, 990 elements of the row will have the value 0. Such a matrix is called a Sparse Matrix.

What Matrix Factorization does is it breaks this matrix (let’s call it A) into lower dimension matrices (Singular Value Decomposition)

PLSA

$$A = L \cdot U \cdot R.$$



The three matrices can be interpreted as —

- L contains the document probabilities P(d|z)
- U is a diagonal matrix of the prior probabilities of the topics P(z)
- R corresponds to the word probability P(w|z)

So if you multiply the three matrices, you actually do what the below equation says

$$P(w|d) = \sum_{z \in Z} P(w|z) P(z|d)$$

Note that the elements of these three matrices cannot be negative as they represent probabilities. Hence, the A matrix is decomposed using Non-Negative Matrix Factorization.

PLSA

Advantages of PLSA

1. It models word-document co-occurrences as a mixture of conditionally independent multinomial distributions.
2. It is considered as a mixture model instead of a clustering model.
3. The results of PLSA have a clear **probabilistic** interpretation.
4. It also allows for model combination.

Disadvantages of PLSA

1. Potentially higher computational complexity.
2. EM algorithm gives local maximum instead of Global Maximum.
3. It is prone to **overfitting**.
4. It is not a well-defined generative model for new documents.

PLSA Demo

- https://www.benfrederickson.com/matrix-factorization

The Expectation-Maximization (EM) Algorithm

- **Expectation-Maximization Algorithm, or EM algorithm for short, is an approach for maximum likelihood estimation in the presence of latent variables.**

- A latent variable model consists of **observable** variables along with **unobservable** variables. Observed variables are those variables in the dataset that can be measured whereas unobserved (latent/hidden) variables are inferred from the observed variables.

- This algorithm is actually at the base of many unsupervised clustering algorithms in the field of machine learning.

The Expectation-Maximization (EM) Algorithm

Algorithm:

1. Given a set of incomplete data, consider a set of starting parameters.
2. Expectation step (E – step): Using the observed available data of the dataset, estimate (guess) the values of the missing data.
3. Maximization step (M – step): Complete data generated after the expectation (E) step is used in order to update the parameters.
4. Repeat step 2 and step 3 until convergence.

The Expectation-Maximization (EM) Algorithm

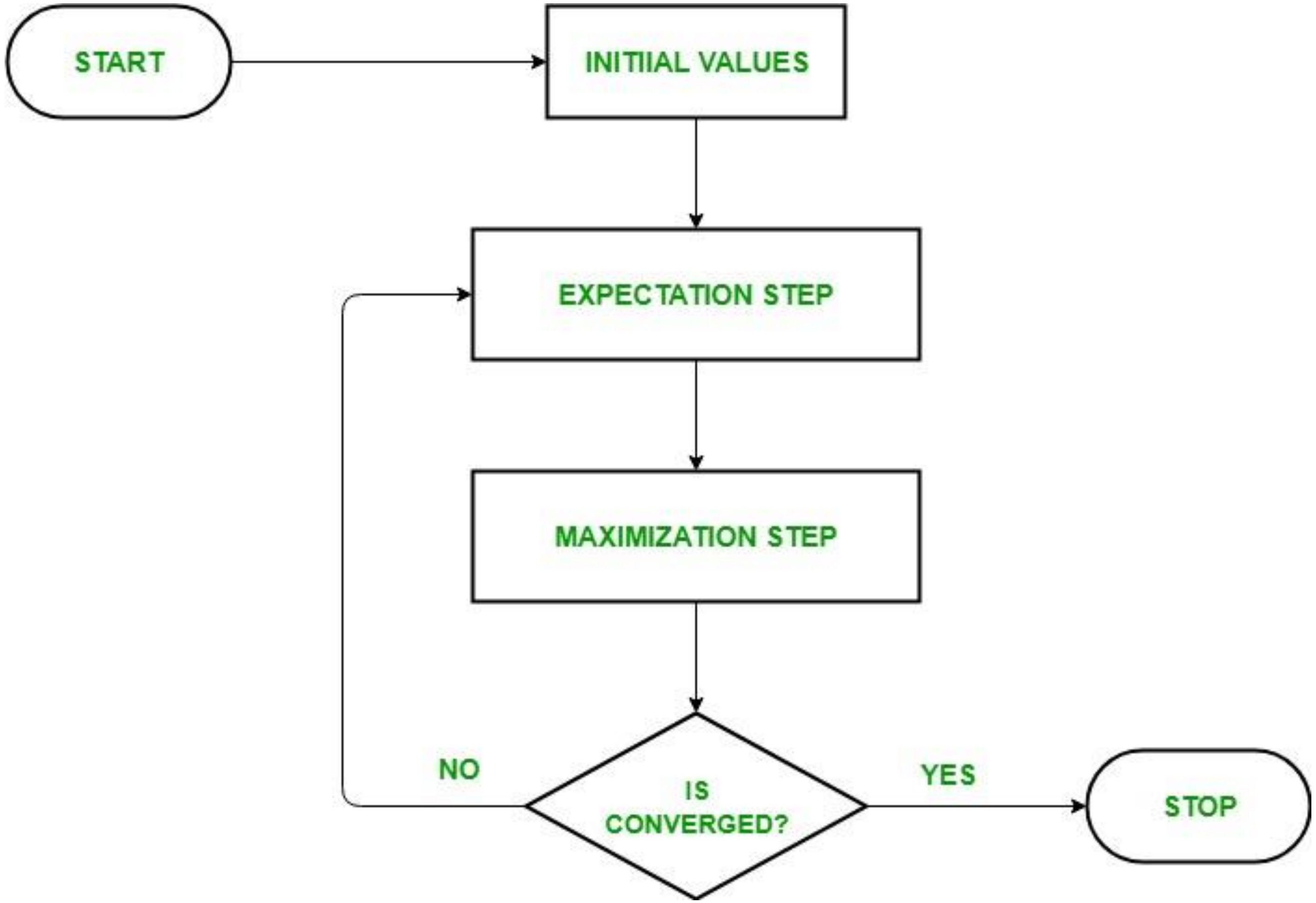
Algorithm:

1. Given a set of incomplete data, consider a set of starting parameters.
2. Expectation step (E – step): Using the observed available data of the dataset, estimate (guess) the values of the missing data.
3. Maximization step (M – step): Complete data generated after the expectation (E) step is used in order to update the parameters.
4. Repeat step 2 and step 3 until convergence.

The essence of Expectation-Maximization algorithm is to use the available observed data of the dataset to estimate the missing data and then using that data to update the values of the parameters.

The Expectation-Maximization (EM) Algorithm

Flow chart for EM algorithm



The Expectation-Maximization (EM) Algorithm

Advantages of EM algorithm:

- It is always guaranteed that likelihood will increase with each iteration.
- The E-step and M-step are often pretty easy for many problems in terms of implementation.
- Solutions to the M-steps often exist in the closed form.

Disadvantages of EM algorithm:

- It has **slow** convergence.
- It makes convergence to the local optima only.

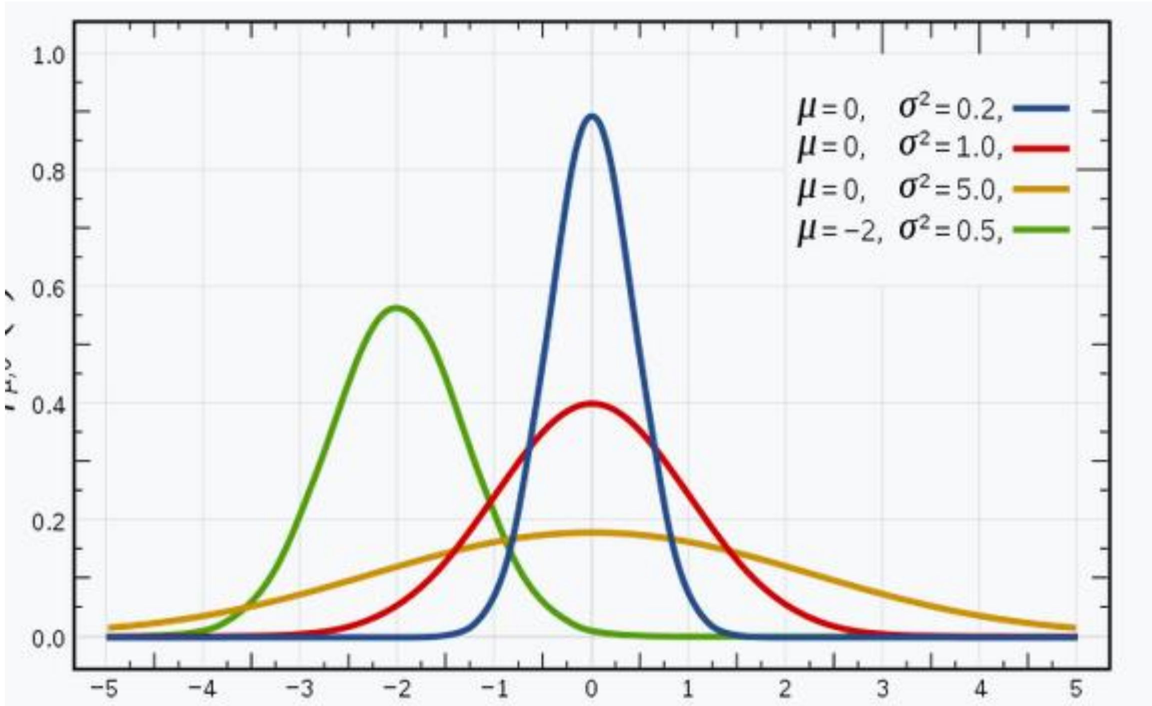
Expectation-Maximization (EM) Algorithm - DEMO

https://colab.research.google.com/drive/1IJCjOG7mwzrKUiyjosdS33mmnqH2X7f-?usp=sharing

Gaussian Distribution

- Gaussian Distributions (or the Normal Distribution) since this distribution is heavily used in the field of Machine Learning and Statistics. It has a bell-shaped curve, with the observations symmetrically distributed around the mean (average) value.

- The given image shown has a few Gaussian distributions with different values of the mean (μ) and variance (σ2). Remember that the higher the σ (standard deviation) value more would be the spread along the axis.



Gaussian Process

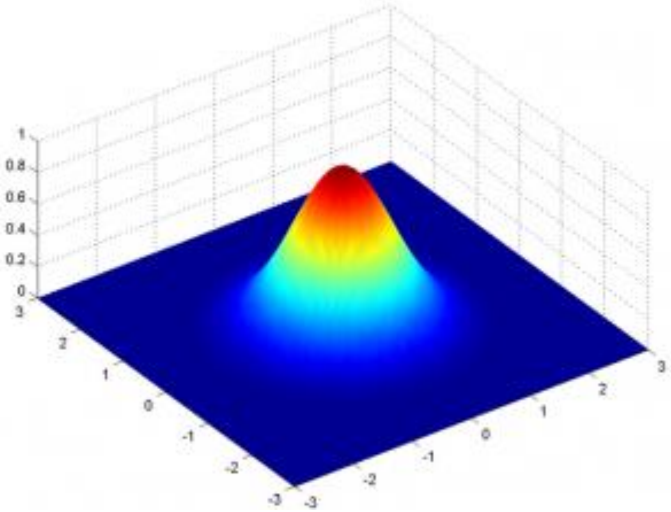
In 1-D space, the probability density function of a Gaussian distribution is given by:

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where μ represents the mean and σ2 represents the variance.

But this would only be true for a variable in 1-D only. In the case of two variables, we will have a 3D bell curve instead of a 2D bell-shaped curve as shown below:

Hence, for a dataset having *d* features, we would have a mixture of *k* Gaussian distributions (where *k* represents the number of clusters), each having a certain mean vector and variance matrix.



End of Unit 3

Thank You.