# The Resistance Notes

3/11

Bayes Rule/Theorem:

Graphical user interface, text, application, email

Description automatically generatedDescribes the probability of an event, based on prior knowledge of conditions that might be related to the event, one of the many applications of Bayes theorem is Bayesian inference, a particular approach to statistical inference. When applied the probabilities involved in the theorem may have different probability interpretations. With Bayesian probability interpretation, the theorem expresses how a degree of belied, expressed as a probability should rationally change to account for the availability of related evidence. Bayesian inference is fundamental to Bayesian statistics.

Bayesian interpretation:

In the Bayesian interpretation, probability measures a “degree of belief”. Bayes’ theorem links the degree of belief in a proposition before and after accounting for evidence. For example, suppose it is believed with 50% certainty that a coin is twice as likely to land heads than tails. If the coin if flipped a number of times and the outcomes observed, that degree of belief will probably rise of fall, but might even remain the same, depending on the results

For proposition *A* and evidence *B*,

* *P* (*A*), the *prior*, is the initial degree of belief in *A*.
* *P* (*A* | *B*), the *posterior*, is the degree of belief after incorporating news that *B* is true.
* the quotient *P*(*B* | *A*)/*P*(*B*) represents the support *B* provides for *A*.

<https://en.wikipedia.org/wiki/Bayes%27_theorem>

Naïve Bayes Classifier Algorithm

* Is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems
* It is mainly used in text classification that includes a high-dimensional training dataset
* Naïve Bayes classifier is one of the simple and most effective classification algorithms which helps in building the fast machine learning models that can make quick predications
* It is a probabilistic classifier, which means it predicts on the basis of the probability of an object

A pretty good link explaining Naïve bayes algorithm with an pretty good example ( not sure if naïve bayes if the way to go yet)

<https://www.javatpoint.com/machine-learning-naive-bayes-classifier>

Bayseian Algorithms

A family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other. Naïve Bayes clssifiers are a collection fo classification algorithms based on Baye’s theorem. Bayes’s formula provides relationship between P(A |B) and P(B|A)

Text

Description automatically generated

Naïve Bayes

A Naïve Bayes algorithm assumes that each of the features it uses are conditionally independent of one another given some class. It provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c).

Some things to consider:

Useful for very large data sets – you can use Naïve Bayes classification algorithm with a small data set but precision and recall will keep very low Since the algorithm has an assumption of independence, you do lose the ability to exploit the interactions between features.

Bayesian Network (BN)

Bayesian networks are a type of Probabilistic Graphical Model (probabilistic because they are built form probability distributions). These networks can be used for predictions, anomaly detection, diagnostics, automated insight, reasoning, time series prediction and **decision making under uncertainty.**

The goal of these networks is to model conditional dependence, and therefore causation. For example: if your outside of your house and it starts raining, there is a high probability that your dog will start barking. Which in turn, will increase the probability that the cat will hide under the couch. So you can see how info about one event (rain) allows you to make inference about a seemingly unrelated event ( the cat hiding under the couch)

Some things to consider:

You can use them to make future predictions

Useful for explaining observations

Bayesian networks are very convenient for representing similar probabilistic relationships between multiple events.

<https://towardsdatascience.com/ml-algorithms-one-sd-%CF%83-bayesian-algorithms-b59785da792a>

“I like the sound of Bayesian Networks gonna dive deeper into them”

Bayesian Network is a probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph (DAG). Bayesian networks are ideal for taking an event that occurred and predicating the likelihood that any one of several possible known causes was the contributing factor.

For example A Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms the network can be used to compute the probabilities of the presence of various diseases

Efficient Algorithms can perform inference and learning in Bayesian networks. Bayesian networks that model sequences of variables are called dynamic Bayesian networks. Generalizations of Bayesian networks that can represent and solve decision problems under uncertainty are called influence diagrams

Formally, Bayesian networks are directed acyclic graphs whose nodes represent variables in the Bayesian sense: they may be observable quantities, latent variables, unknown parameters or hypotheses. Edges represent conditional dependencies; nodes that are not connected (no path connects on node to another) represent variables that are conditionally independent of each other. Each node is associated with a probability function that takes, as input, a particular set of values for the nodes parent variables, and gives (as output) the probability of the variable represented by the node.

For example, if m parent nodes represent m Boolean variables, the probability function could be represented by a 2m entries, one entry for each of the 2m possible parent combinations.

Inferring unobserved variables

Because a Bayesian network is a complete model for its variables and their relationships it can be used to answer probabilistic queries about them. For example, the network can be used to update knowledge of the state of a subset of variables when other variables ( the evidence variables) are observed. This process of computing the posterior distribution of variables given evidence is called probabilistic inference. The posterior gives a universal sufficient statistic for detection applications, when choosing values for the variable subset that minimize some expected loss function, for instance the probability of decision error. A Bayesian network can thus be considered a mechanism for automatically applying Bayes theorem to complex problems

The most common exact inference methods are: *variable elimination*, which eliminates (by integration or summation) the non-observed non-query variables one by one by distributing the sum over the product;

*Clique tree propagation:* which caches the computation so that many variables can be queried at one time and new evidence can be propagated quickly;

And *recursive conditioning and AND/OR search,* which allow for a space-time tradeoff and match the efficiency of variable elimination when enough space is used.

All of these methods have complexity that is exponential in the networks treewidth. The most common approximate inference algorithms are *importance sampling, stochastic MCMC simulation, mini-bucket elimination, loopy belief propagation, generalized belief propagation and variational methods.*

[*https://en.wikipedia.org/wiki/Bayesian\_network*](https://en.wikipedia.org/wiki/Bayesian_network)

*“I like the sound of variable elimination going to dive deeper”*

Variable elimination

Graphical user interface, text, application

Description automatically generatedVariable elimination (VE) is a simple and general exact inference algorithm in probabilistic graphical models, such as Bayesian networks and Markov random files. It can used for inference of maximum a posteriori (MAP) state or estimation of conditional o marginal distributions over a subset of variables. The algorithm has exponential time complexity, but could be efficient in practice for the low-treewidth graphs, if the proper elimination order is used

Graphical user interface, text, application, email

Description automatically generatedOrdering

Finding the optimal order in which to eliminate variables is an NP-hard problem. As such there are heuristic one may follow to better optimize performance by order:

1. *Minimum Degree:* eliminate the variable which results in constructing the smallest factor possible
2. *Minimum Fill:* By constructing an undirected graph showing variable relations expressed by all CPTs, eliminate the variable which would result in the lease edges to be assed post elimination

<https://en.wikipedia.org/wiki/Variable_elimination>

https://en.wikipedia.org/wiki/Minimum\_degree\_algorithm

Really good Bayesian networks link with examples and code snippets:

https://www.edureka.co/blog/bayesian-networks/

Bayesian networks code looks a bit scary, maybe naïve bayes classifiers will be a bit easier

<https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/>

<https://www.analyticsvidhya.com/blog/2021/01/a-guide-to-the-naive-bayes-algorithm/>

<https://www.youtube.com/watch?v=PPeaRc-r1OI>Table

Description automatically generated

Text

Description automatically generated

A picture containing text, blackboard

Description automatically generated

<https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/>

<https://boardgames.stackexchange.com/questions/4719/optimal-or-just-effective-strategy-for-the-resistance>

good description of usual gameplay by user Todd^

*lets talk tactics:*

Resistance

* Always pick yourself to go on missions
* If previous missions were successful. Pick the same team and potentially add another if required
  + Only spies would generally have cause to vote against this team
* If a team leader doesn’t go for the same team again that was mission success they are more likely a spy

Chart, scatter chart

Description automatically generatedChart

Description automatically generated

Chart

Description automatically generatedChart

Description automatically generatedChart

Description automatically generatedChart, histogram

Description automatically generated

New 9/10

Chart

Description automatically generatedChart

Description automatically generatedChart

Description automatically generatedChart, histogram

Description automatically generatedChart

Description automatically generated

NEW

LOGICAL RESISTANCE VS LOGICAL SPY AGENTS (NO BAYES CLASSIFIER)

Resistance Wins Vs Spy Wins 113 , 387

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generatedChart

Description automatically generatedChart, histogram

Description automatically generatedChart, histogram

Description automatically generatedChart, histogram

Description automatically generatedChart

Description automatically generatedChart

Description automatically generatedGraphical user interface

Description automatically generated with medium confidenceChart, histogram

Description automatically generated

LOGICAL RESISTANCE VS RANDOM SPY AGENTS (NO TRAINING DATA)

Resistance Wins Vs Spy Wins 199 , 23

Chart, histogram

Description automatically generatedGraphical user interface, application

Description automatically generatedChart, histogram

Description automatically generatedChart

Description automatically generatedGraphical user interface, application

Description automatically generatedChart

Description automatically generatedGraphical user interface, text, application

Description automatically generatedChart, histogram

Description automatically generatedChart

Description automatically generatedChart

Description automatically generated