Demand Forecasting and Master Algorithms

Moon Hyunji 2017_08

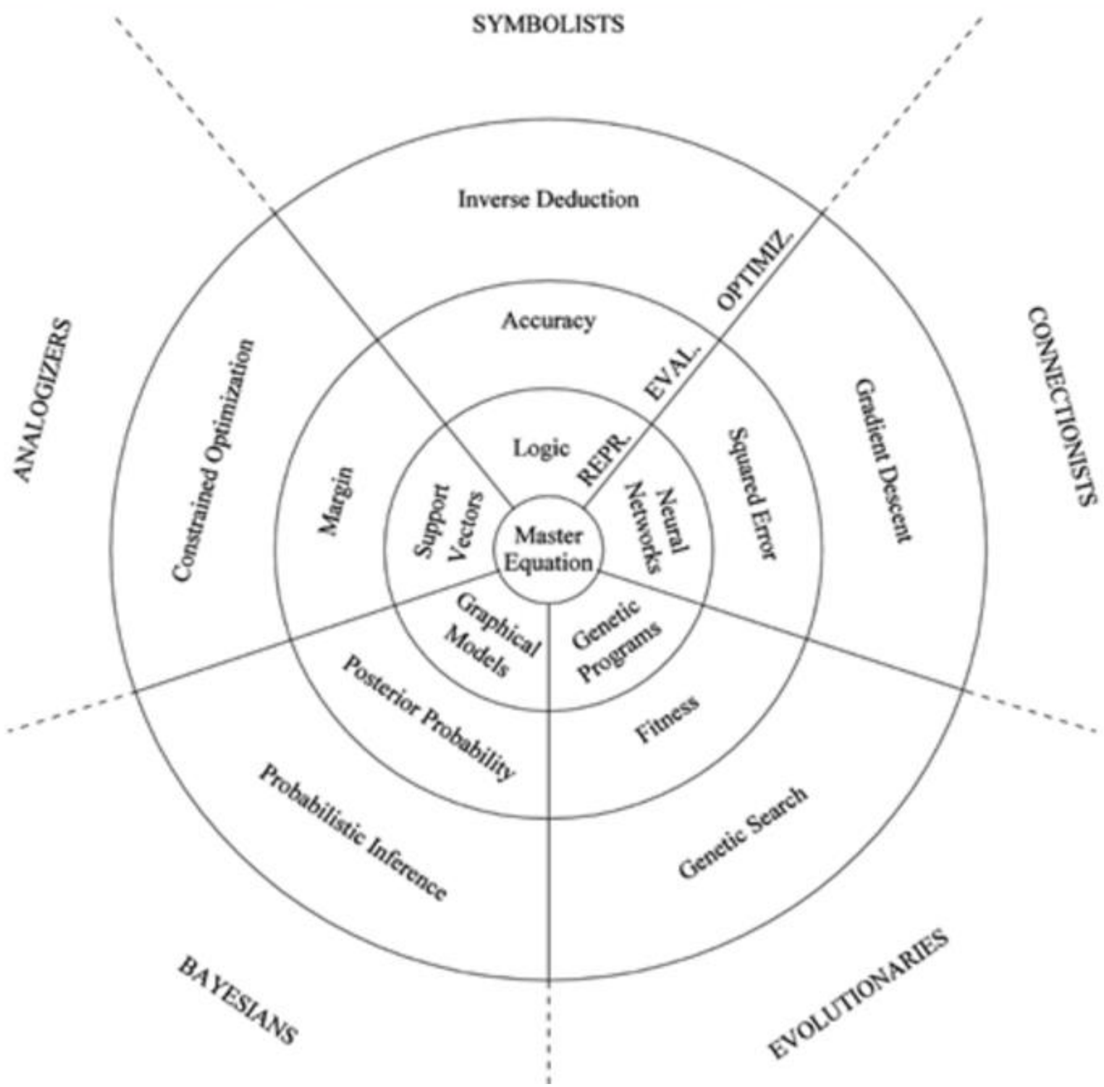
INDEX

Master algorithm

- ***Symbolists***
- ***Connectionists***
- ***Evolutionaries***
- ***Bayesians***
- ***Analogizers***

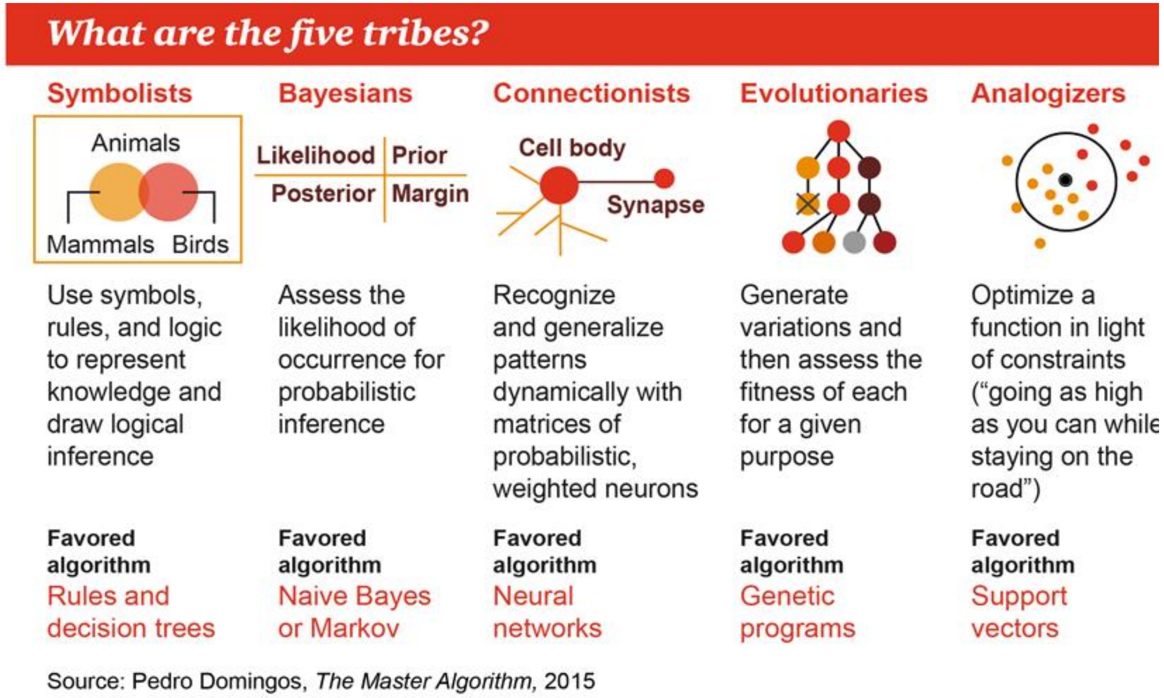
Moon Hyunji 2017_08

Tribe	Origins	Master Algorithm
Symbolists	Logic, philosophy	Inverse deduction
Connectionists	Neuroscience	Backpropagation
Evolutionaries	Evolutionary biology	Genetic programming
Bayesians	Statistics	Probabilistic inference
Analogizers	Psychology	Kernel machines

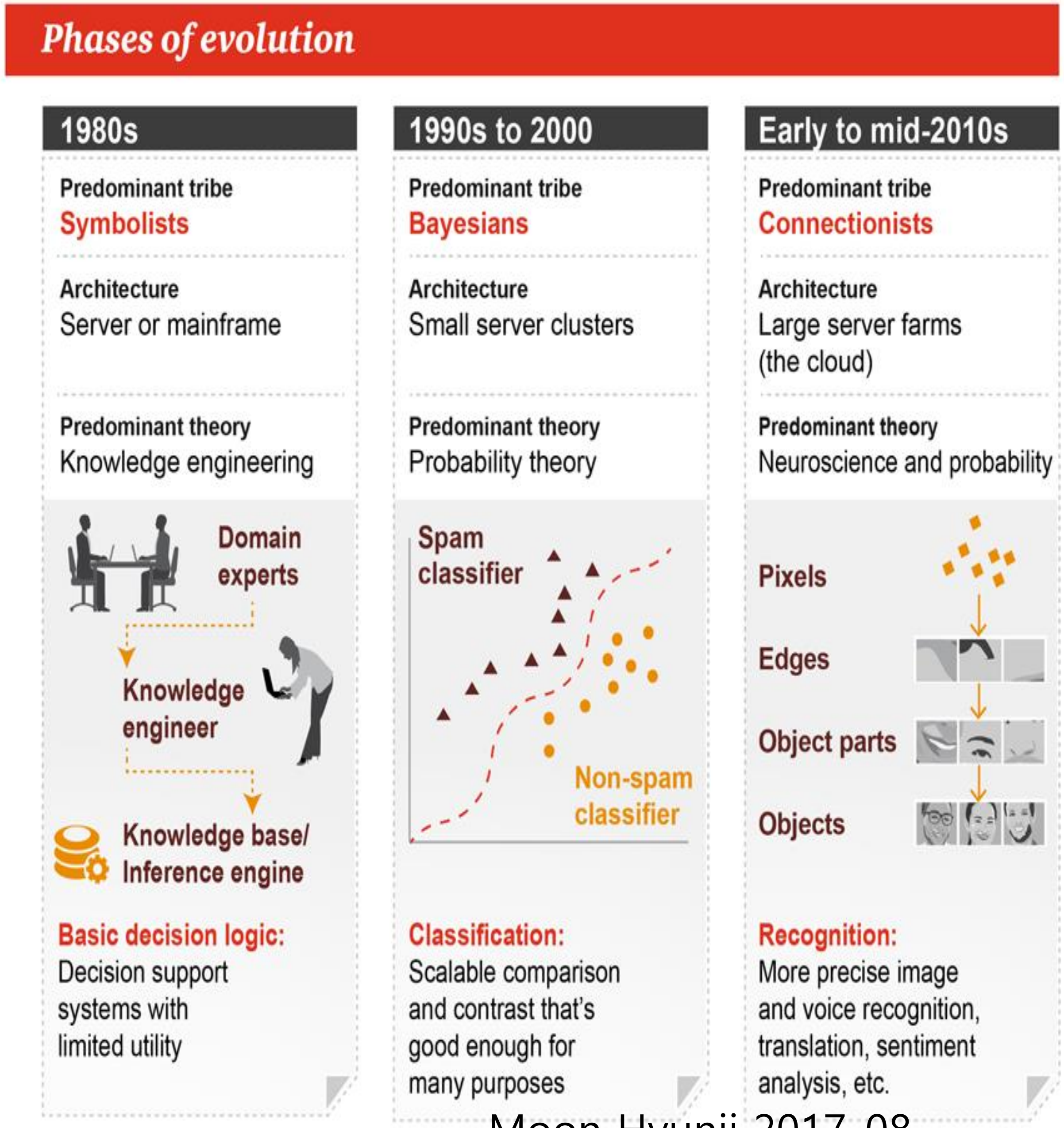


Moon Hyunji 2017_08

CONTENTS. 01 Introduction



- Machine learning evolution
  http://usblogs.pwc.com/emerging-technology/machine-learning-evolution-infographic/
- Machine learning methods
  http://usblogs.pwc.com/emerging-technology/machine-learning-methods-infographic/
- Machine learning overview
  http://usblogs.pwc.com/emerging-technology/a-look-at-machine-learning-infographic/

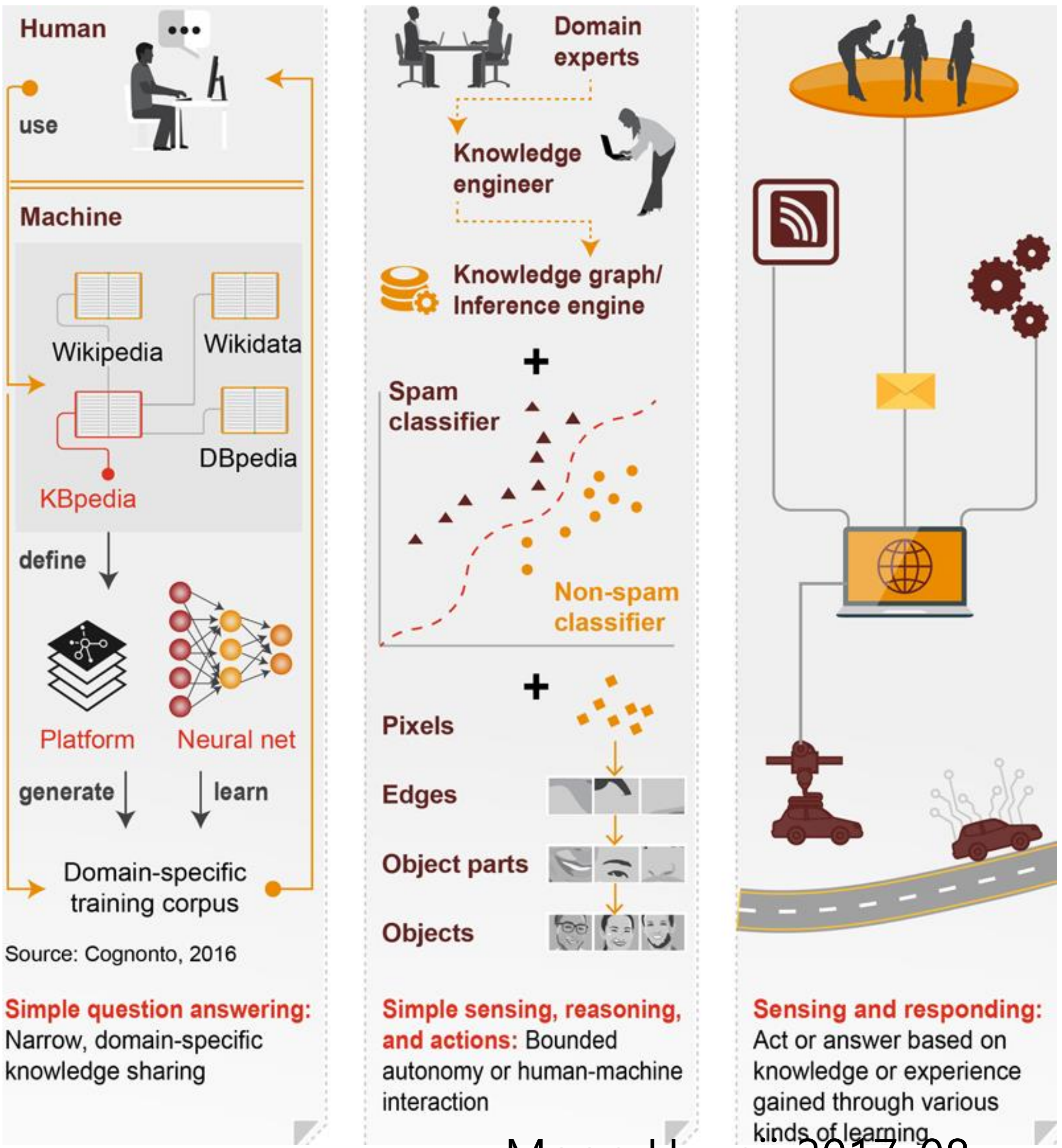


Moon Hyunji 2017_08

CONTENTS. 01 Introduction

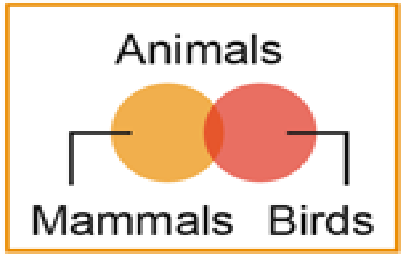
The tribes see fit to collaborate and blend their methods

Late 2010s	2020s+	2040s+
Predominant tribe **Connectionists + Symbolists**	Predominant tribe **Connectionists + Symbolists + Bayesians + ...**	Predominant tribe **Algorithmic convergence**
Architecture Multiple clouds	Architecture Clouds and fog	Architecture Server ubiquity
Predominant theory Memory neural networks, large-scale integration, and reasoning over knowledge	Predominant theory Networks when sensing, but rules when reasoning and acting	Predominant theory Best-of-breed meta-learning



Moon Hyunji 2017_08

2.1 Symbolists



Use symbols, rules, and logic to represent knowledge and draw logical inference

Favored algorithm
Rules and decision trees

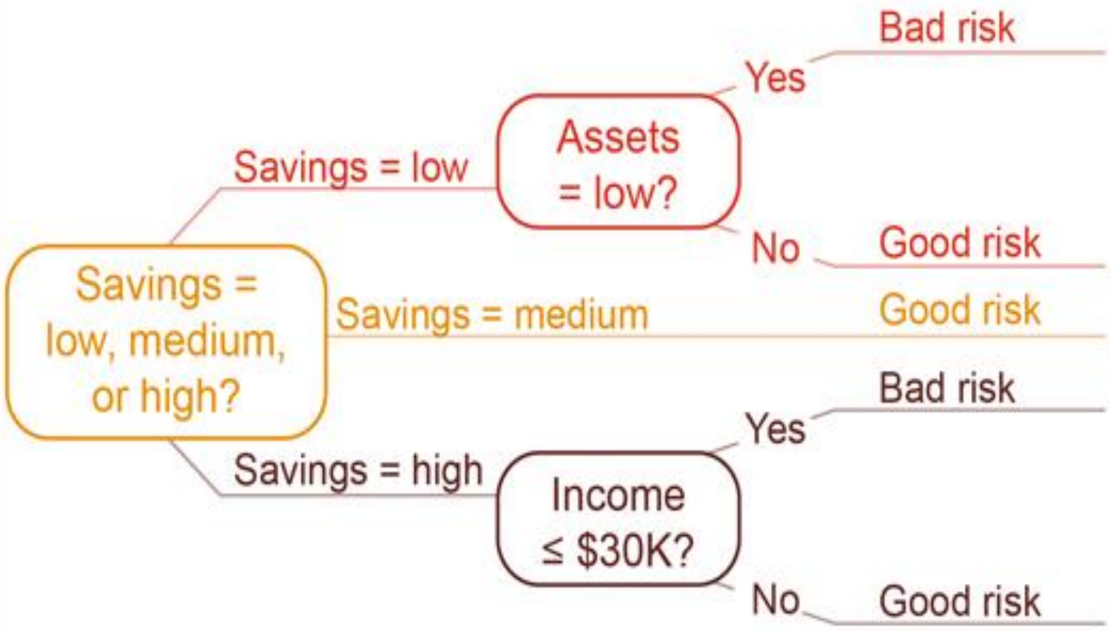
Use symbols, rules, and logic to represent knowledge and draw logical inference
view learning as the inverse of deduction

- key algorithm: inverse deduction: figures out what knowledge is missing in order to make a deduction go through, and then makes it as general as possible
- key to learning: how to incorporate preexisting knowledge into learning, and how to combine different pieces of knowledge to solve new problems
- all intelligence can be reduced to manipulating symbols, in the same way that a mathematician solves equations by replacing expressions by other expressions

2.1 Symbolists

Decision trees

Decision tree analysis typically uses a hierarchy of variables or decision nodes that, when answered step by step, can classify a given customer as creditworthy or not, for example.

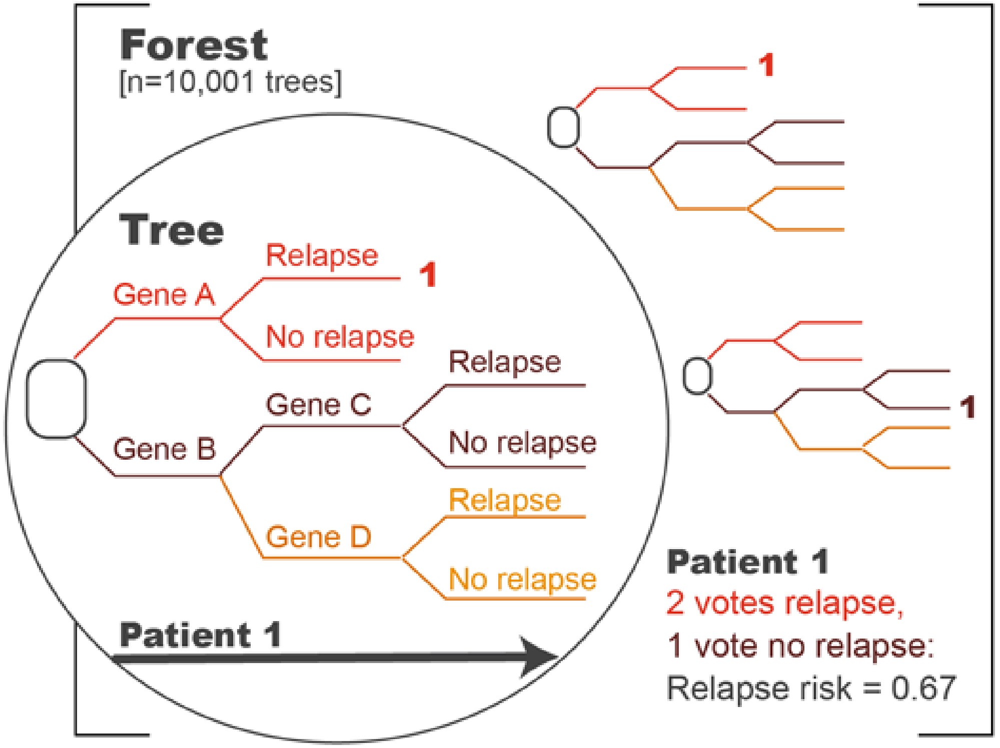


Advantages	Use cases
Decision trees are useful when evaluating lists of distinct features, qualities, or characteristics of people, places, or things.	Rule-based credit risk assessment, horse race performance prediction

Source: Daniel T. Larose and Chantal D. Larose, *Data Mining and Predictive Analytics*, 2nd Edition, John Wiley & Sons, 2015

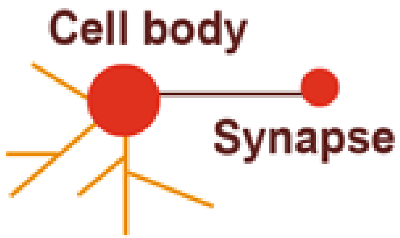
Random forest

Random forest algorithms improve the accuracy of decision trees by using multiple trees with randomly selected subsets of data. This example reviews the expression levels of various genes associated with breast cancer relapse and computes a relapse risk.



Advantages	Use cases
Random forest methods prove useful with large data sets and items that have numerous and sometimes irrelevant features.	Customer churn analysis, risk assessment

Source: Nicolas Spies, Washington University, 2015



Recognize and generalize patterns dynamically with matrices of probabilistic, weighted neurons

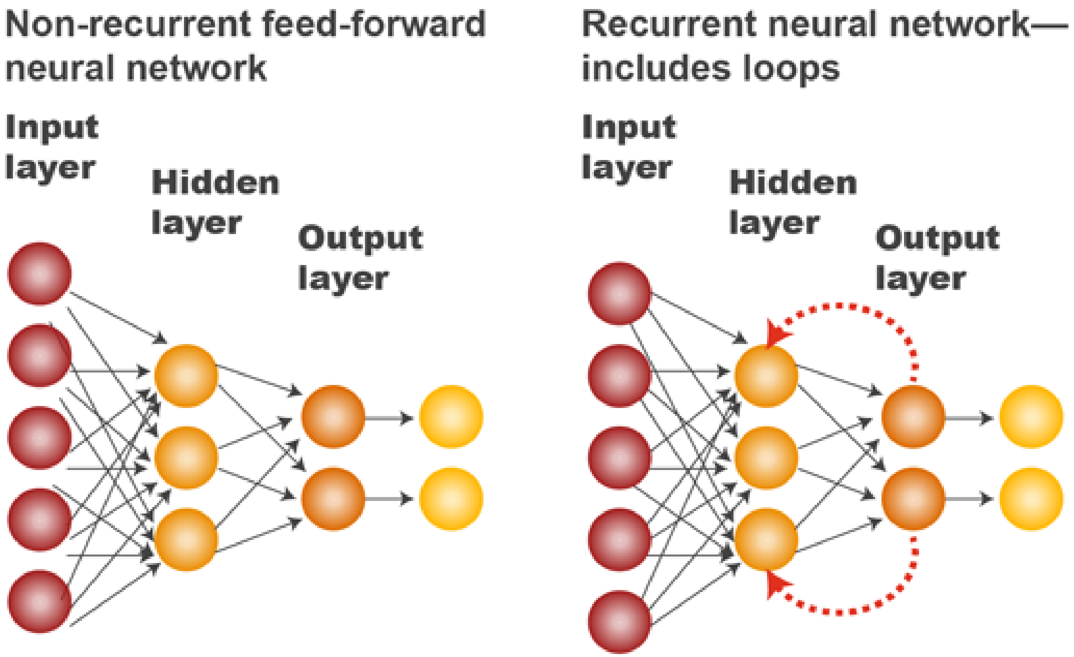
Favored algorithm
Neural networks

reverse engineer the brain and are inspired by neuroscience and physics

- key algorithm: backpropagation
- key to learning: learning is what the brain does, and so what we need to do is reverse engineer it
- brain learns by adjusting the strengths of connections between neurons, and the crucial problem is figuring out which connections are to blame for which errors and changing them accordingly (= backpropagation)

Recurrent neural networks

Each neuron in any neural network converts many inputs into single outputs via one or more hidden layers. Recurrent neural networks [RNNs] additionally pass values from step to step, making step-by-step learning possible. In other words, RNNs have a form of memory, allowing previous outputs to affect subsequent inputs.

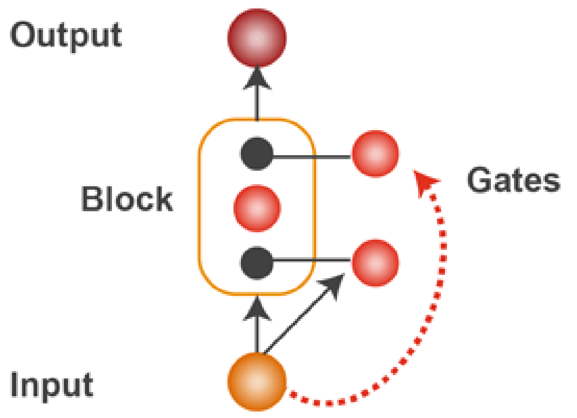


Advantages	Use cases
Recurrent neural networks have predictive power when used with large amounts of sequenced information.	Image classification and captioning, political sentiment analysis

Source: Joseph Wilks, 2012

Long short-term memory & gated recurrent unit neural networks

Older forms of RNNs can be lossy. While these older recurrent neural networks only allow small amounts of older information to persist, newer long short-term memory (LSTM) and gated recurrent unit (GRU) neural networks have both long- and short-term memory. In other words, these newer RNNs have greater memory control, allowing previous values to persist or to be reset as necessary for many sequences of steps, avoiding "gradient decay" or eventual degradation of the values passed from step to step. LSTM and GRU networks make this memory control possible with memory blocks and structures called gates that pass or reset values as appropriate.



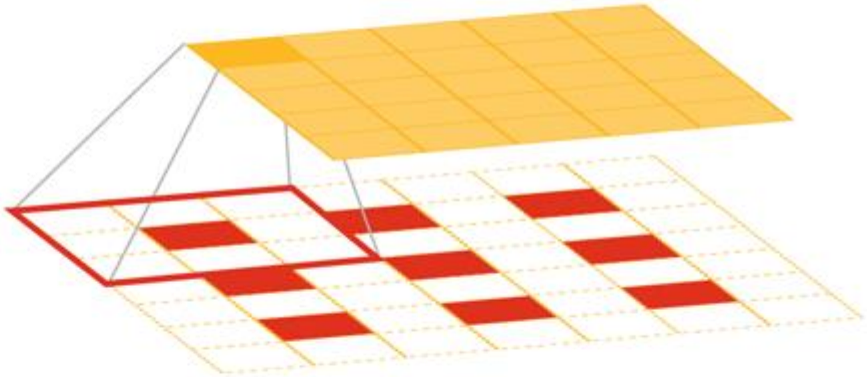
Source: Genevieve Orr, et al., Williamette University, 1999

Advantages	Use cases
Long short-term memory and gated recurrent unit neural networks have the same advantages as other recurrent neural networks and are more frequently used than other recurrent neural networks because of their greater memory capabilities.	Natural language processing, translation

2.2 Connectionist

Convolutional neural networks

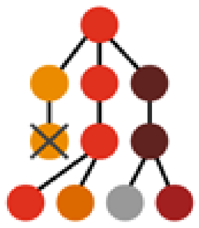
Convolutions are blends of weights from a subsequent layer that are used to label the output layer.



Advantages	Use cases
Convolutional neural networks are most useful with very large data sets, large numbers of features, and complex classification tasks.	Image recognition, text to speech, drug discovery

Source: Algobeans, 2016

2.3 Evolutionaries



Generate variations and then assess the fitness of each for a given purpose

Favored algorithm
Genetic programs

simulate evolution on the computer and draw on genetics and evolutionary biology.

- key algorithm: genetic programming: mates and evolves computer programs in the same way that nature mates and evolves organisms.
- key to learning: learning structure: not just adjusting parameters, like backpropagation does, but creating the brain that those adjustments can then fine-tune

2.3 Evolutionaries

- Genetic Algorithm: finding optimality in an agent-based way
- used in NP problems such as TSP

- In computer science and operations research, a **genetic algorithm** (**GA**) is a metaheuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms (EA). Genetic algorithms are commonly used to generate high-quality solutions to optimization and search problems by relying on bio-inspired operators such as mutation, crossover and selection.[1] -wikipedia

2.4 Bayesians

Likelihood	Prior
Posterior	Margin

Assess the likelihood of occurrence for probabilistic inference

Favored algorithm
Naive Bayes or Markov

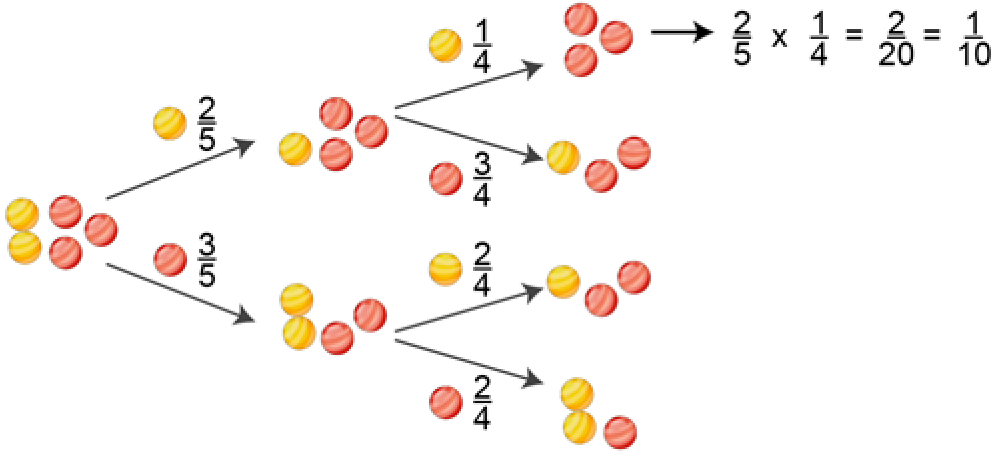
believe learning is a form of probabilistic inference and have their roots in statistics.

- key algorithm: Bayesian inference
- key to learning: how to incorporate new evidence into our beliefs, and probabilistic inference algorithm do that efficiently as possible
- All learned knowledge is uncertain, and learning itself is a form of uncertain inference
- how to deal with noisy, incomplete, and even contradictory information without falling apart→ probabilistic inference

2.4 Bayesians

Naive Bayes classification

Naive Bayes classifiers compute probabilities, given tree branches of possible conditions. Each individual feature is "naive" or conditionally independent of, and therefore does not influence, the others. For example, what's the probability you would draw two yellow marbles in a row, given a jar of five yellow and red marbles total? The probability, following the topmost branch of two yellow in a row, is one in ten. Naive Bayes classifiers compute the combined, conditional probabilities of multiple attributes.



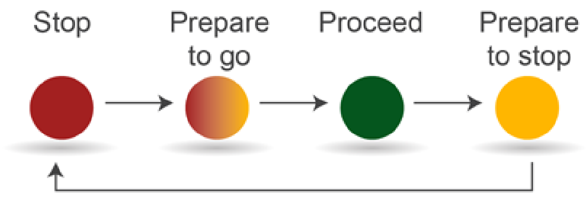
Advantages	Use cases
Naive Bayes methods allow the quick classification of relevant items in small data sets that have distinct features.	Sentiment analysis, consumer segmentation

Source: Rod Pierce, et al., *MathIsFun*, 2014

2.4 Bayesians

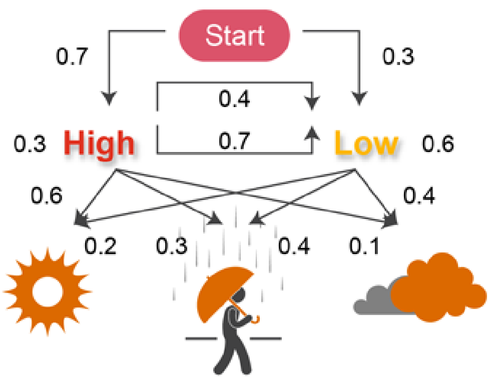
Hidden Markov models

Observable Markov processes are purely deterministic—one given state always follows another given state. Traffic light patterns are an example.



Source: Derek Kane, 2015

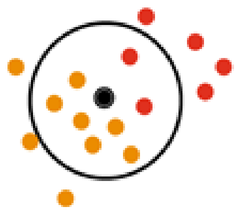
Hidden Markov models, by contrast, compute the probability of hidden states occurring by analyzing observable data, and then estimating the likely pattern of future observation with the help of the hidden state analysis. In this example, the probability of high or low pressure (the hidden state) is used to predict the likelihood of sunny, rainy, or cloudy weather.



Advantages	Use cases
Tolerates data variability and effective for recognition and prediction.	Facial expression analysis, weather prediction

Source: Leonardo Guizzetti, 2012

2.5 Analogizers



Optimize a function in light of constraints ("going as high as you can while staying on the road")

Favored algorithm
Support vectors

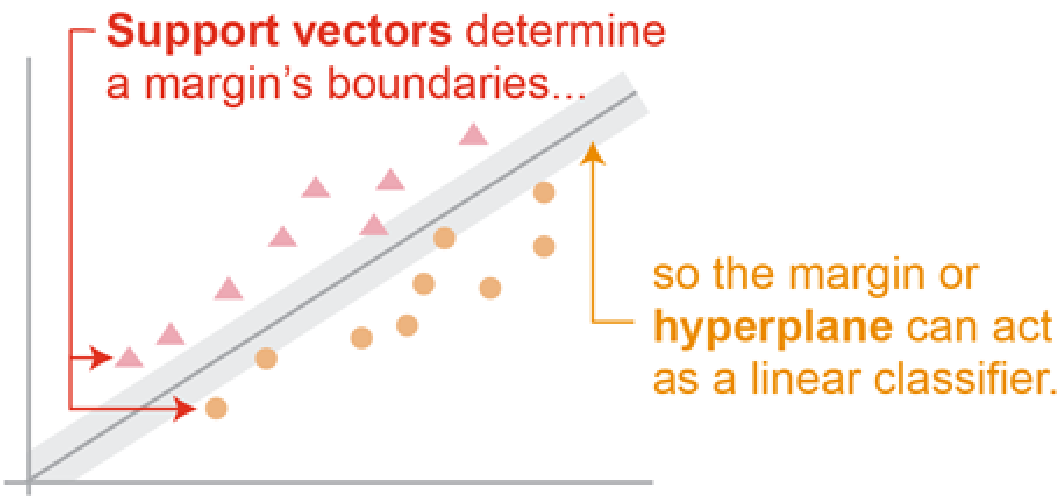
learn by extrapolating from similarity judgments and are influenced by psychology and mathematical optimization.

- key algorithm: support vector machine: figures out which experiences to remember and how to combine them to make new predictions
- key to learning: recognizing similarities between situations and thereby inferring other similarities, judging how similar two things are

2.5 Analogizers

Support vector machines

Support vector machines classify groups of data with the help of hyperplanes.

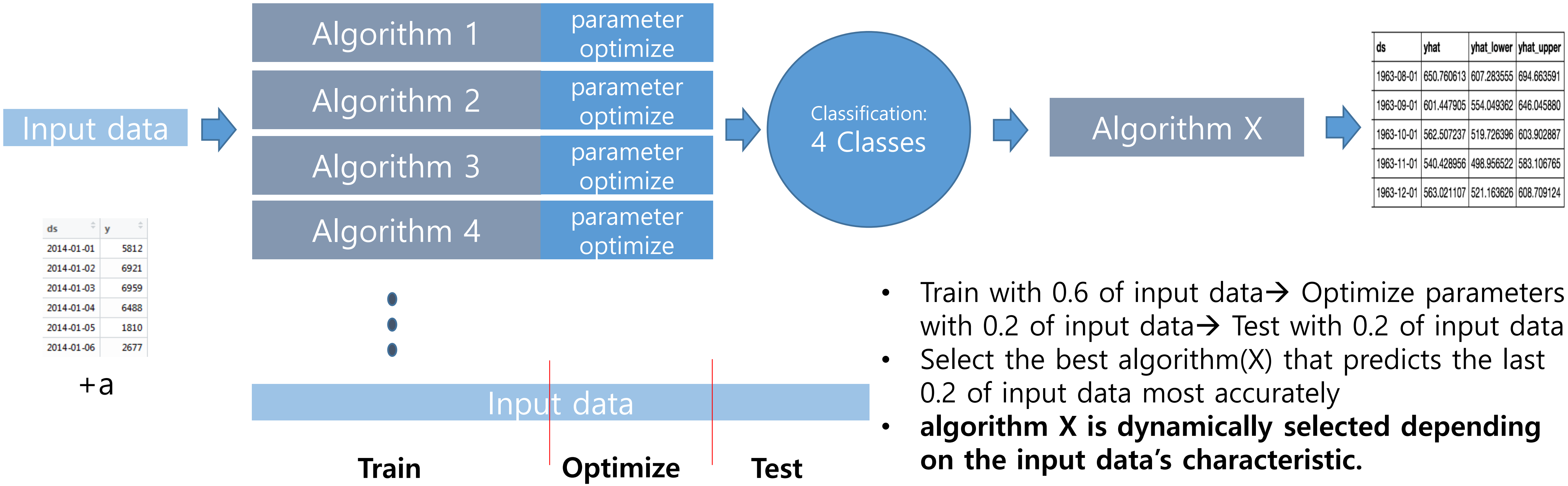


Advantages	Use cases
Support vector machines are good for the binary classification of X versus other variables and are useful whether or not the relationship between variables is linear.	News categorization, handwriting recognition

Source: Matthew Kelly, *Computer Science: Source*, 2010

2.6 **Ensemble (proposal)**

- Framework design1



ds	y
2014-01-01	5812
2014-01-02	6921
2014-01-03	6959
2014-01-04	6488
2014-01-05	1810
2014-01-06	2677

ds	yhat	yhat_lower	yhat_upper
1963-08-01	650.760613	607.283555	694.663591
1963-09-01	601.447905	554.049362	646.045880
1963-10-01	562.507237	519.726396	603.902887
1963-11-01	540.428956	498.956522	583.106765
1963-12-01	563.021107	521.163626	608.709124

- Train with 0.6 of input data→ Optimize parameters with 0.2 of input data→ Test with 0.2 of input data
- Select the best algorithm(X) that predicts the last 0.2 of input data most accurately
- **algorithm X is dynamically selected depending on the input data's characteristic.**

Moon Hyunji 2017_08

2.7 **Hypothesis**

Designing a high accuracy learning system

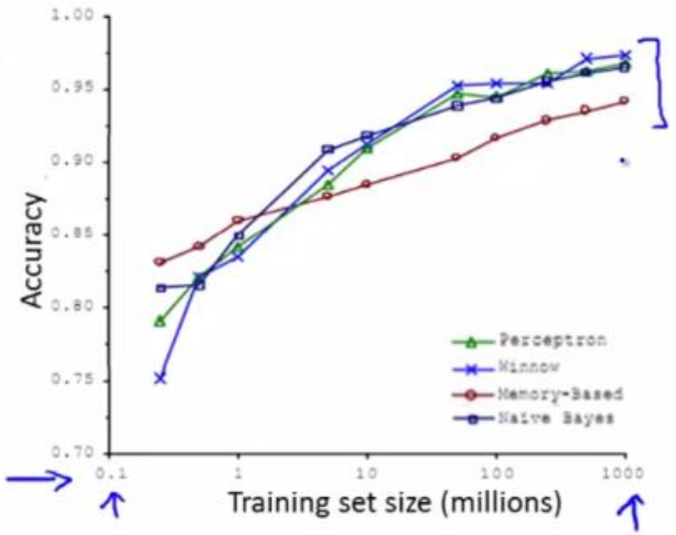
E.g. Classify between confusable words.

{to, two, too}, {then, than}

For breakfast I ate two eggs.

Algorithms

- Perceptron (Logistic regression)
- Winnow
- Memory-based
- Naïve Bayes



"It's not who has the best algorithm that wins.

It's who has the most data."

[Banko and Brill, 2001]

3. Algorithms for Time Series

TBC