

[Start Here](#)

[Blog](#)

[Training](#)

[About](#)

A Tour of Machine Learning Algorithms

by **Jason Brownlee** on November 25, 2013 in **Uncategorized**

877

1931

227

992

In this post we take a tour of the most popular machine learning algorithms. It is useful to tour the main algorithms in the field to get a feeling of what methods are available.

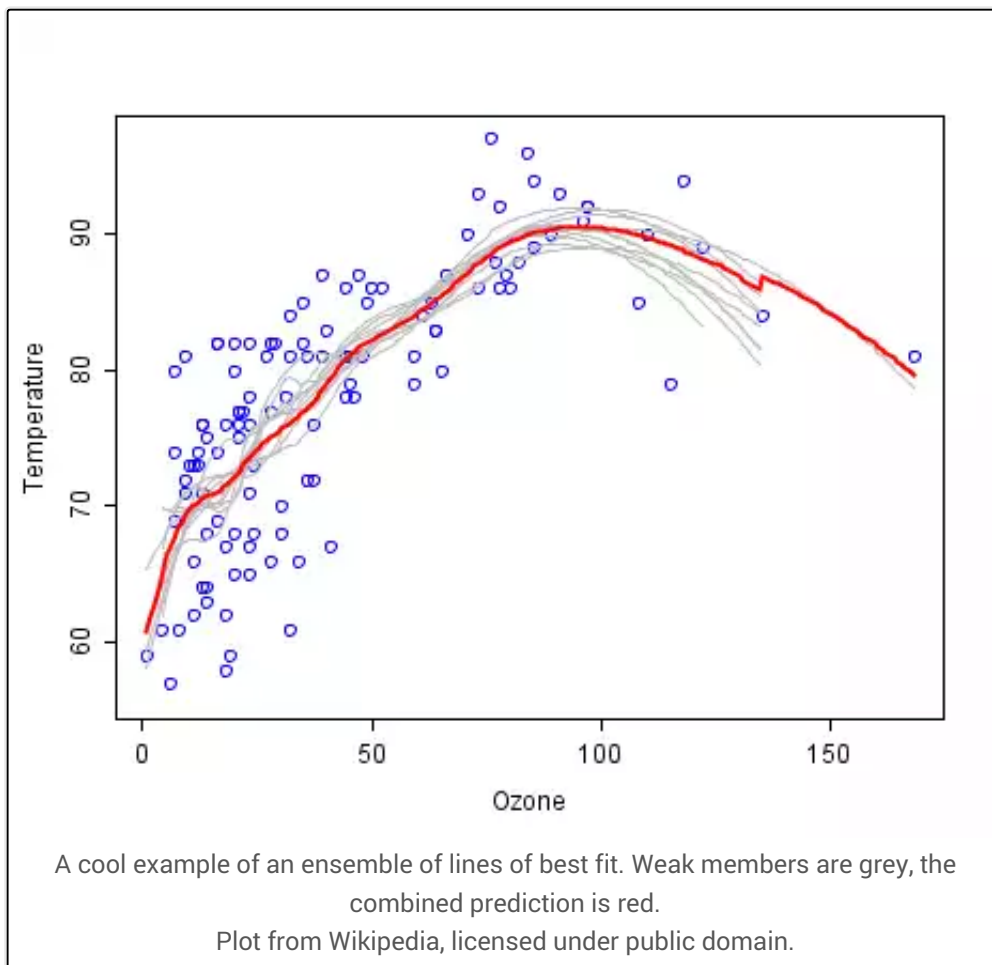
There are so many algorithms available and it can feel overwhelming when algorithm names are thrown around and you are expected to just know what they are and where they fit.

In this post I want to give you two ways to think about and categorize the algorithms you may come across in the field.

- The first is a grouping of algorithms by the **learning style**.
- The second is a grouping of algorithms by **similarity** in form or function (like grouping similar animals together).

Both approaches are useful, but we will focus in on the grouping of algorithms by similarity and go on a tour of a variety of different algorithm types.

After reading this post, you will have a much better understanding of the most popular machine learning algorithms for supervised learning and how they are related.



Algorithms Grouped by Learning Style

There are different ways an algorithm can model a problem based on its interaction with the experience or environment or whatever we want to call the input data.

It is popular in machine learning and artificial intelligence textbooks to first consider the learning styles that an algorithm can adopt.

There are only a few main learning styles or learning models that an algorithm can have and we'll go through them here with a few examples of algorithms and problem types that they suit.

This taxonomy or way of organizing machine learning algorithms is useful because it forces you to think about the roles of the input data and the model preparation process and select one that is the most appropriate for your problem in order to get the best result.

Let's take a look at four different learning styles in machine learning algorithms:

Supervised Learning

Input data is called training data and has a known label or result such as spam/not-spam or a stock price at a time.

A model is prepared through a training process where it is required to make predictions and is corrected when those predictions are wrong. The training process continues until the model

achieves a desired level of accuracy on the training data.

Example problems are classification and regression.

Example algorithms include Logistic Regression and the Back Propagation Neural Network.

Unsupervised Learning

Input data is not labelled and does not have a known result.

A model is prepared by deducing structures present in the input data.

This may be to extract general rules. It may through a mathematical

process to systematically reduce redundancy, or it may be to organize data by similarity.

Example problems are clustering, dimensionality reduction and association rule learning.

Example algorithms include: the Apriori algorithm and k-Means.

Semi-Supervised Learning

Input data is a mixture of labelled and unlabelled examples.

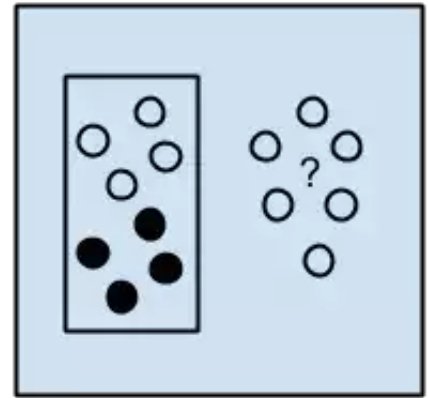
There is a desired prediction problem but the model must learn the structures to organize the data as well as make predictions.

Example problems are classification and regression.

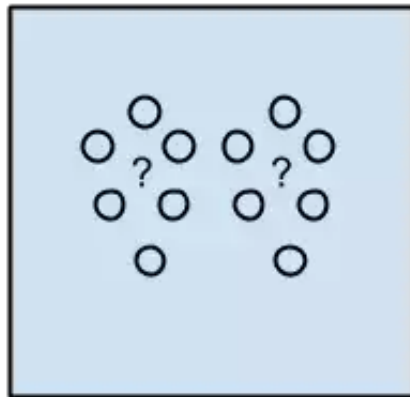
Example algorithms are extensions to other flexible methods that make assumptions about how to model the unlabelled data.

Overview

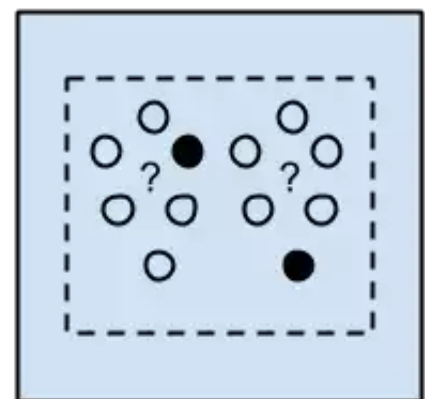
When crunching data to model business decisions, you are most typically using supervised and



Supervised Learning Algorithms



Unsupervised Learning Algorithms



Semi-supervised Learning Algorithms

unsupervised learning methods.

A hot topic at the moment is semi-supervised learning methods in areas such as image classification where there are large datasets with very few labelled examples.

Algorithms Grouped By Similarity

Algorithms are often grouped by similarity in terms of their function (how they work). For example, tree-based methods, and neural network inspired methods.

I think this is the most useful way to group algorithms and it is the approach we will use here.

This is a useful grouping method, but it is not perfect. There are still algorithms that could just as easily fit into multiple categories like Learning Vector Quantization that is both a neural network inspired method and an instance-based method. There are also categories that have the same name that describes the problem and the class of algorithm such as Regression and Clustering.

We could handle these cases by listing algorithms twice or by selecting the group that subjectively is the “best” fit. I like this latter approach of not duplicating algorithms to keep things simple.

In this section I list many of the popular machine learning algorithms grouped the way I think is the most intuitive. It is not exhaustive in either the groups or the algorithms, but I think it is representative and will be useful to you to get an idea of the lay of the land.

Please Note: There is a strong bias towards algorithms used for classification and regression, the two most prevalent supervised machine learning problems you will encounter.

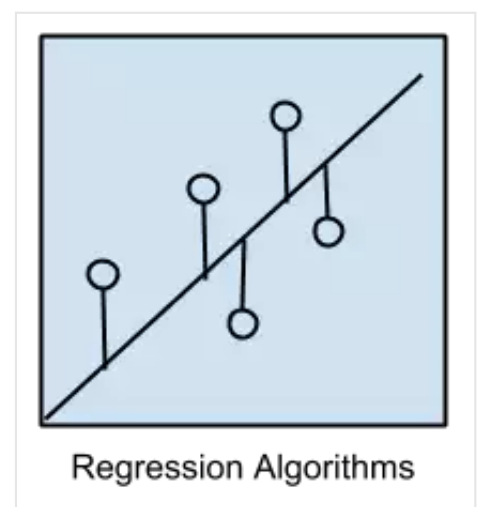
If you know of an algorithm or a group of algorithms not listed, put it in the comments and share it with us. Let's dive in.

Regression Algorithms

Regression is concerned with modelling the relationship between variables that is iteratively refined using a measure of error in the predictions made by the model.

Regression methods are a workhorse of statistics and have been cooped into statistical machine learning. This may be confusing because we can use regression to refer to the class of problem and the class of algorithm. Really, regression is a process.

The most popular regression algorithms are:



- Ordinary Least Squares Regression (OLSR)
- Linear Regression
- Logistic Regression
- Stepwise Regression
- Multivariate Adaptive Regression Splines (MARS)
- Locally Estimated Scatterplot Smoothing (LOESS)

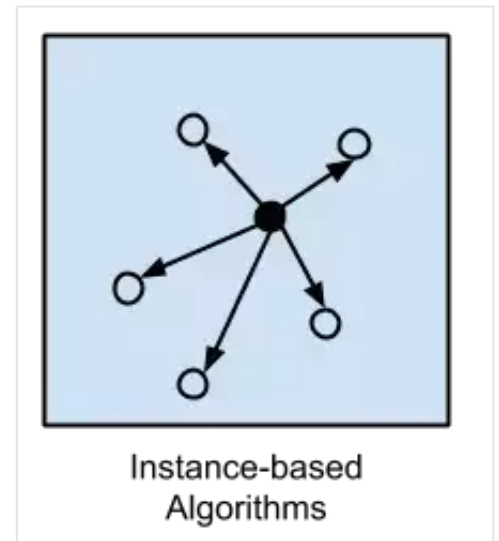
Instance-based Algorithms

Instance based learning model a decision problem with instances or examples of training data that are deemed important or required to the model.

Such methods typically build up a database of example data and compare new data to the database using a similarity measure in order to find the best match and make a prediction. For this reason, instance-based methods are also called winner-take-all methods and memory-based learning. Focus is put on representation of the stored instances and similarity measures used between instances.

The most popular instance-based algorithms are:

- k-Nearest Neighbour (kNN)
- Learning Vector Quantization (LVQ)
- Self-Organizing Map (SOM)
- Locally Weighted Learning (LWL)



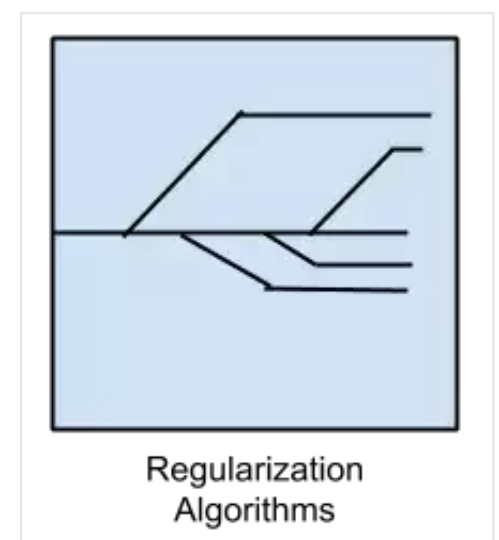
Regularization Algorithms

An extension made to another method (typically regression methods) that penalizes models based on their complexity, favoring simpler models that are also better at generalizing.

I have listed regularization algorithms separately here because they are popular, powerful and generally simple modifications made to other methods.

The most popular regularization algorithms are:

- Ridge Regression
- Least Absolute Shrinkage and Selection Operator (LASSO)
- Elastic Net
- Least-Angle Regression (LARS)



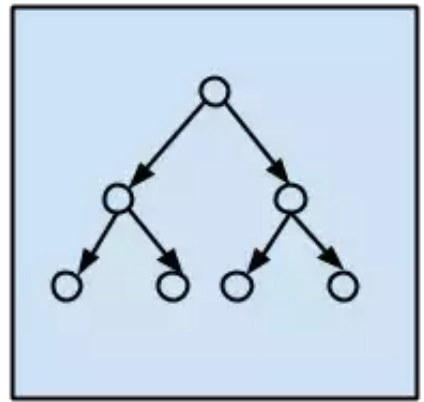
Decision Tree Algorithms

Decision tree methods construct a model of decisions made based on actual values of attributes in the data.

Decisions fork in tree structures until a prediction decision is made for a given record. Decision trees are trained on data for classification and regression problems. Decision trees are often fast and accurate and a big favorite in machine learning.

The most popular decision tree algorithms are:

- Classification and Regression Tree (CART)
- Iterative Dichotomiser 3 (ID3)
- C4.5 and C5.0 (different versions of a powerful approach)
- Chi-squared Automatic Interaction Detection (CHAID)
- Decision Stump
- M5
- Conditional Decision Trees



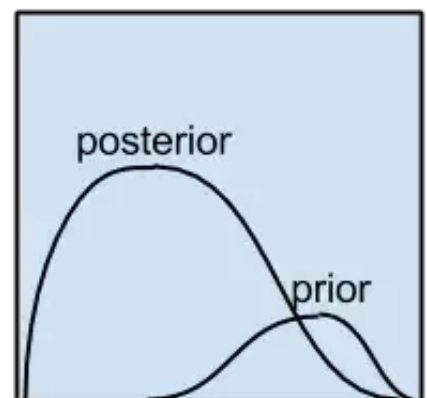
Decision Tree Algorithms

Bayesian Algorithms

Bayesian methods are those that explicitly apply Bayes' Theorem for problems such as classification and regression.

The most popular Bayesian algorithms are:

- Naive Bayes
- Gaussian Naive Bayes
- Multinomial Naive Bayes
- Averaged One-Dependence Estimators (AODE)
- Bayesian Belief Network (BBN)
- Bayesian Network (BN)



Bayesian Algorithms

Clustering Algorithms

Clustering, like regression describes the class of problem and the class of methods.

Clustering methods are typically organized by the modelling approaches such as centroid-based and hierarchal. All methods are concerned with using the inherent structures in the data to best organize the data into groups of maximum commonality.

The most popular clustering algorithms are:

- k-Means
- k-Medians
- Expectation Maximisation (EM)
- Hierarchical Clustering

Association Rule Learning Algorithms

Association rule learning are methods that extract rules that best explain observed relationships between variables in data.

These rules can discover important and commercially useful associations in large multidimensional datasets that can be exploited by an organisation.

The most popular association rule learning algorithms are:

- Apriori algorithm
- Eclat algorithm

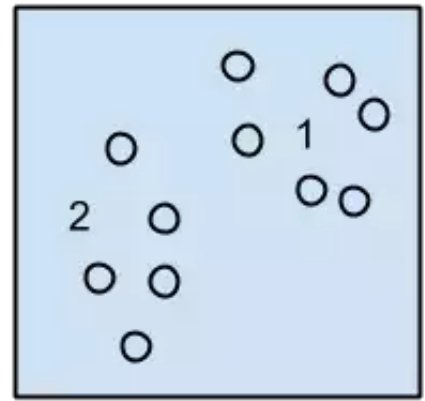
Artificial Neural Network Algorithms

Artificial Neural Networks are models that are inspired by the structure and/or function of biological neural networks.

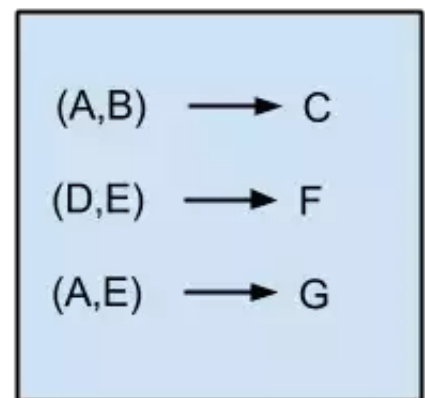
They are a class of pattern matching that are commonly used for regression and classification problems but are really an enormous subfield comprised of hundreds of algorithms and variations for all manner of problem types.

Note that I have separated out Deep Learning from neural networks because of the massive growth and popularity in the field. Here we are concerned with the more classical methods.

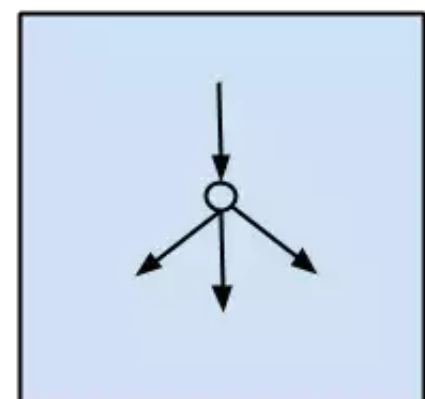
The most popular artificial neural network algorithms are:



Clustering Algorithms



Association Rule Learning Algorithms



Artificial Neural Network Algorithms

- Perceptron
- Back-Propagation
- Hopfield Network
- Radial Basis Function Network (RBFN)

Deep Learning Algorithms

Deep Learning methods are a modern update to Artificial Neural Networks that exploit abundant cheap computation.

They are concerned with building much larger and more complex neural networks, and as commented above, many methods are concerned with semi-supervised learning problems where large datasets contain very little labelled data.

The most popular deep learning algorithms are:

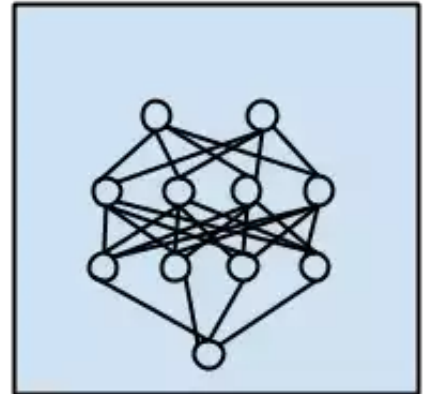
- Deep Boltzmann Machine (DBM)
- Deep Belief Networks (DBN)
- Convolutional Neural Network (CNN)
- Stacked Auto-Encoders

Dimensionality Reduction Algorithms

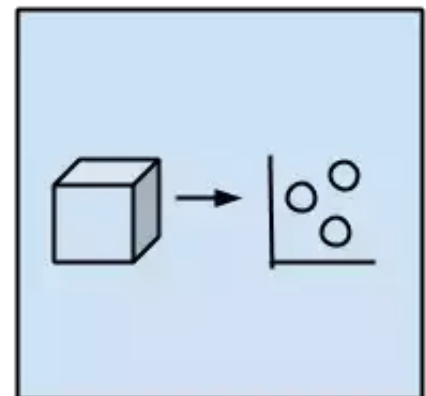
Like clustering methods, dimensionality reduction seek and exploit the inherent structure in the data, but in this case in an unsupervised manner or order to summarise or describe data using less information.

This can be useful to visualize dimensional data or to simplify data which can then be used in a supervised learning method. Many of these methods can be adapted for use in classification and regression.

- Principal Component Analysis (PCA)
- Principal Component Regression (PCR)
- Partial Least Squares Regression (PLSR)
- Sammon Mapping
- Multidimensional Scaling (MDS)
- Projection Pursuit
- Linear Discriminant Analysis (LDA)
- Mixture Discriminant Analysis (MDA)
- Quadratic Discriminant Analysis (QDA)
- Flexible Discriminant Analysis (FDA)



Deep Learning Algorithms



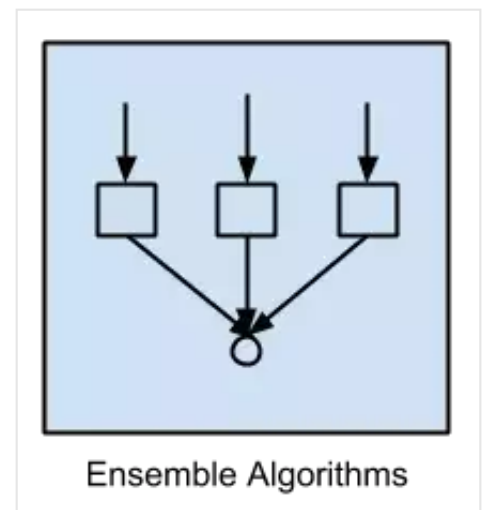
Dimensional Reduction Algorithms

Ensemble Algorithms

Ensemble methods are models composed of multiple weaker models that are independently trained and whose predictions are combined in some way to make the overall prediction.

Much effort is put into what types of weak learners to combine and the ways in which to combine them. This is a very powerful class of techniques and as such is very popular.

- Boosting
- Bootstrapped Aggregation (Bagging)
- AdaBoost
- Stacked Generalization (blending)
- Gradient Boosting Machines (GBM)
- Gradient Boosted Regression Trees (GBRT)
- Random Forest



Other Algorithms

Many algorithms were not covered.

For example, what group would Support Vector Machines go into? It's own?

I did not cover algorithms from speciality tasks in the process of machine learning, such as:

- Feature selection algorithms
- Algorithm accuracy evaluation
- Performance measures

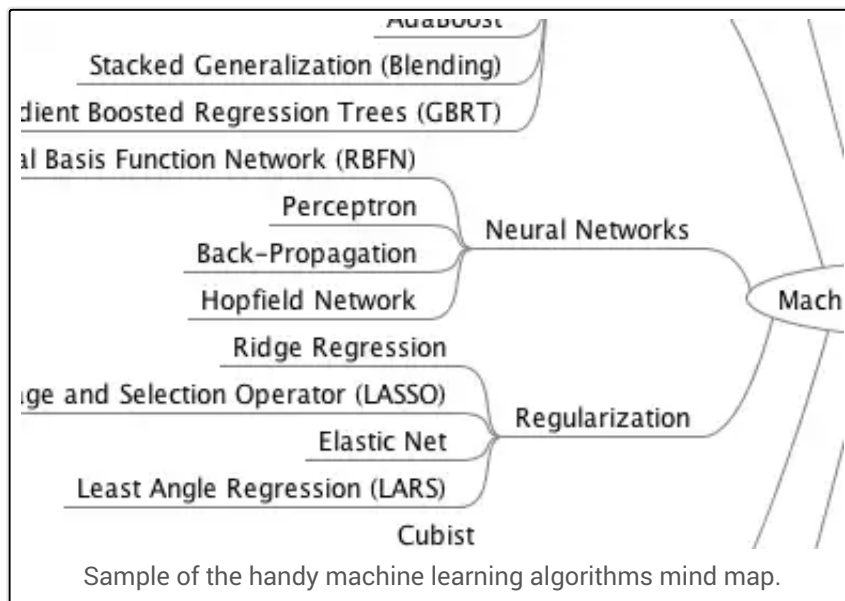
I also did not cover algorithms from speciality sub-fields of machine learning, such as:

- Computational intelligence (evolutionary algorithms, etc.)
- Computer Vision (CV)
- Natural Language Processing (NLP)
- Recommender Systems
- Reinforcement Learning
- Graphical Models
- And more...

These may feature in future posts.

Get your FREE Algorithms Mind Map

I've created a handy mind map of 60+ algorithms organized by type.



Download it, print it and use it to jump-start your next machine learning project.

[Download For Free](#)

Also receive exclusive email tips and tricks.

Further Reading

This tour of machine learning algorithms was intended to give you an overview of what is out there and some ideas on how to relate algorithms to each other.

I've collected together some resources for you to continue your reading on algorithms. If you have a specific question, please leave a comment.

Other Lists of Algorithms

There are other great lists of algorithms out there if you're interested. Below are few hand selected examples.

- [List of Machine Learning Algorithms](#): On Wikipedia. Although extensive, I do not find this list or the organization of the algorithms particularly useful.
- [Machine Learning Algorithms Category](#): Also on Wikipedia, slightly more useful than Wikipedias great list above. It organizes algorithms alphabetically.
- [CRAN Task View: Machine Learning & Statistical Learning](#): A list of all the packages and all the algorithms supported by each machine learning package in R. Gives you a grounded feeling of what's out there and what people are using for analysis day-to-day.
- [Top 10 Algorithms in Data Mining: Published article](#) and now a [book](#) (Affiliate Link) on the most popular algorithms for data mining. Another grounded and less overwhelming take on methods that you could go off and learn deeply.

How to Study Machine Learning Algorithms

Algorithms are a big part of machine learning. It's a topic I am passionate about and write about a lot on this blog. Below are few hand selected posts that might interest you for further reading.

- [How to Learn Any Machine Learning Algorithm](#): A systematic approach that you can use to study and understand any machine learning algorithm using "algorithm description templates" (I used this approach to write [my first book](#)).
- [How to Create Targeted Lists of Machine Learning Algorithms](#): How you can create your

own systematic lists of machine learning algorithms to jump start work on your next machine learning problem.

- [How to Research a Machine Learning Algorithm](#): A systematic approach that you can use to research machine learning algorithms (works great in collaboration with the template approach listed above).
- [How to Investigate Machine Learning Algorithm Behavior](#): A methodology you can use to understand how machine learning algorithms work by creating and executing very small studies into their behaviour. Research is not just for academics!
- [How to Implement a Machine Learning Algorithm](#): A process and tips and tricks for implementing machine learning algorithms from scratch.

How to Run Machine Learning Algorithms

Sometimes you just want to dive into code. Below are some links you can use to run machine learning algorithms, code them up using standard libraries or implement them from scratch.

- [How To Get Started With Machine Learning Algorithms in R](#): Links to a large number of code examples on this site demonstrating machine learning algorithms in R.
- [Machine Learning Algorithm Recipes in scikit-learn](#): A collection of Python code examples demonstrating how to create predictive models using scikit-learn.
- [How to Run Your First Classifier in Weka](#): A tutorial for running your very first classifier in Weka (**no code required!**).

Final Word

I hope you have found this tour useful.

Please, leave a comment if you have any questions or ideas on how to improve the algorithm tour.

Update #1: Continue the [discussion on HackerNews](#) and [reddit](#).

Update #2: I've added a bunch more resources and more algorithms. I've also added a handy mind map that you can download (see above).



About Jason Brownlee

The editor-in-chief at MachineLearningMastery.com. Jason is a husband, father, researcher, author, professional programmer and a machine learning practitioner. [Learn more about him.](#)

[View all posts by Jason Brownlee](#) →

56 Responses to *A Tour of Machine Learning Algorithms*



Bruce December 20, 2013 at 5:10 pm #

REPLY ↩

What about reinforcement learning algorithms in algorithm similarity classification?
There is also one called Gibbs algorithm under Bayesian Learning



jasonb December 26, 2013 at 8:34 pm #

REPLY ↩

Good point bruce, I left out those methods. Would you like me to write a post about reinforcement learning methods?



Jason's fan August 22, 2015 at 6:39 am #

REPLY ↩

Machine Learning Crash Course

Machine Learning Crash Course



You can learn and apply Machine Learning even without a background in maths or a fancy degree. Find out how in this ridiculously practical (and totally free) email course.

SIGN ME UP

small datasets using a tool like Weka. It's a GUI tool and provides a bunch of standard datasets and algorithms out of the box.

I'd suggest you build up some skill on small datasets before moving onto big data tools like Hadoop and Mahout.



swainjo June 9, 2014 at 6:24 pm #

REPLY ↩

qnaguru,

I would recommend the Coursera courses.

I would also read a couple of books to give you some background into the possibilities and limitations. Nate Silver; The Signal and The Noise & Danial Kahneman; Thinking Fast and Slow.



Ismo May 20, 2014 at 2:50 am #

REPLY ↩

The best written one I have found is: "The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition". However you probably need to have some background on maths/stats/computing before reading that (especially if you are planning to implement them too). For general algorithms implementation I recommend reading also "Numerical Recipes 3rd Edition: The Art of Scientific Computing".



jasonb May 23, 2014 at 8:01 am #

REPLY ↩

I'm a huge fan of Numerical Recipes, thanks for the book refs.



William May 23, 2014 at 1:37 am #

REPLY ↩

Not a single one for recommender systems?



jasonb May 23, 2014 at 8:02 am #

REPLY ↩

I would call recommender a higher-order system that internally is solving regression or classification problems. Do you agree?



Jon May 23, 2014 at 2:47 am #

REPLY ↩

genetic algorithms seem to be dying a slow death these days (discussed previously <https://news.ycombinator.com/item?id=7712824>)



jasonb May 23, 2014 at 8:01 am #

REPLY ↩

It's a good point. Computers are fast enough that you can enumerate the search space faster than a GA can converge (at least with the classical toy problems).



Alex May 23, 2014 at 8:47 am #

REPLY ↩

I enjoyed this post but I think that this is a misinformed statement. Genetic Algorithms are most useful in large search spaces (enumerating here would be impossible, were talking about spaces that could be 10^{100}) and highly complex non-convex functions. Modern algorithms are much more sophisticated than the simple techniques used in the 80s e.g. (<http://en.wikipedia.org/wiki/CMA-ES>) and (http://en.wikipedia.org/wiki/Estimation_of_distribution_algorithm). Here is a nice fun recent application:
<http://www.cc.gatech.edu/~jtan34/project/learningBicycleStunts.html>



Jason Brownlee August 22, 2015 at 4:39 pm #

REPLY ↩

Thanks Alex, you can also check out my 2011 book of algorithm recipes titled [Clever Algorithms: Nature-Inspired Programming Recipes](#). In it I cover 5 different estimation of distribution algorithms and 10 different evolutionary algorithms.



Vinícius May 23, 2014 at 6:29 am #

REPLY ↩

Hi guys, this is great! What about recommendation systems? I'm fascinated about, how netflix, amazon and others websites can recommend items based on my taste.



jasonb May 23, 2014 at 8:00 am #

REPLY ↩

Good point.
You can break a recommender down into a classification or regression problem.



Rixi July 12, 2014 at 10:52 am #

REPLY ↩

True, or even use rule induction like Apriori...



mycall May 26, 2014 at 3:50 pm #

REPLY ↩

Where does imagination lie? Would it be a Unsupervised Feedback Learning? Maybe its Neural Deep Ensemble Networks. I presume dreaming = imagination while sleeping, hence daydreaming is imagining of new learning algorithms 😊



Jason Brownlee August 22, 2015 at 4:41 pm #

REPLY ↩

This is too deep for me @mycall



vas May 27, 2014 at 5:28 am #

REPLY ↩

I lot of people swear by this chart for helping you narrow down which machine learning approach to take: http://scikit-learn.org/stable/_static/ml_map.png. It doesn't seem to cover all the types you list in your article. Perhaps a more thorough chart would be useful.



Jason Brownlee August 22, 2015 at 4:41 pm #

REPLY ↩

Thanks for the link vas!



Nevil Nayak May 27, 2014 at 7:22 am #

REPLY ↩

Thid is great. I had always been looking for "all types" of ML algorithms available. I enjoyed reading this and look forward to further reading



Jason Brownlee August 22, 2015 at 4:41 pm #

REPLY ↩

You're very welcome @Nevil.



UD May 30, 2014 at 12:42 am #

REPLY ↩

This is nice and useful...I have been feeling heady with too much data and this kinda gives me a menu from which to choose what all is on offer to help me make sense of stuff 😊
Thanks



Jason Brownlee August 22, 2015 at 4:43 pm #

REPLY ↩

That is a great way to think about @UD, a menu of algorithms.

Tim Browning May 30, 2014 at 4:15 am #

REPLY ↩



You might want to include entropy-based methods in your summary. I use relative-entropy based monitoring in my work to identify anomalies in time series data. This approach has a better recall rate and lower false positive rates when tested with synthetic data using injected outliers. Just an idea, your summary is excellent for such a high level conceptual overview.



Bhaskar January 9, 2015 at 7:27 am #

REPLY ↩

Hi Tim

Can you give me some reference from which I can learn about relative-entropy based monitoring ?



Jason Brownlee August 22, 2015 at 4:44 pm #

REPLY ↩

Thanks @Tim, I'll add a section on time series algorithms I think.



Vincent June 9, 2014 at 7:50 pm #

REPLY ↩

Hi,

Thank's for this tour, it is very useful ! But I disagree with you for the LDA method, which is in the Kernel Methods. First of all, by LDA, do you mean Linear Discriminant Analysis ? Because if it's not, the next parts of my comment are useless :p

If you are talking about this method, then you should put KLDA (which stand for Kernel LDA) and not simply LDA. Because LDA is more a dimension reduction method than a kernel method (It finds the best hyperplane that optimize the Fisher discriminant in order to project data on it).

Next, I don't know if we can view the RBF as a real machine learning method, it's more a mapping function I think, but it is clearly used for mapping to a higher dimension.

Except these two points, the post is awesome ! Thank's again.



Jason Brownlee August 22, 2015 at 4:45 pm #

REPLY ↩

Thanks @Vincent, I'll look into moving the algorithms around a bit in their groupings.



Rémi June 10, 2014 at 8:50 pm #

REPLY ↩

Great post, but I agree with Vincent. Kernel Methods are not machine learning methods by themselves, but more an extension that allows to overcome some difficulties encountered when input data are not linearly separable. SVM and LDA are not Kernel-based, but their definition can be adapted to make use of the famous kernel-trick, giving birth to KSVM and KLDA, that are able to separate data linearly in a higher-dimensional space. Kernel trick can be applied to a wide variety of Machine learning methods:

- LDA
- SVM
- PCA
- KMeans

and the list goes on...

Moreover, I don't think that RBF can be considered a machine learning method. It is a kernel function used alongside the kernel trick to project the data in a high-dimensional space. So the listing in "Kernel methods" seems to have a typing error :p

Last point, don't you think LDA could be added to the "Dimensionality Reduction" category ? In fact, it's more an open question but, mixture methods (clustering) and factor analysis could be considered "Dimensionality Reduction methods" since data can be labeled either by its cluster id, or its factors.

Thanks again for this post, giving an overview of machine learning methods is a great thing.



Jason Brownlee August 22, 2015 at 4:47 pm #

REPLY ↩

Great comments @Rémi I'll move things around a bit.



Pranav Waila June 10, 2014 at 9:24 pm #

REPLY ↩

Hi qnaguru, I have collected some nice reference books to start digging Machine learning. I would suggest you to start with "Introduction to statistical learning" and after that you can look into "The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition", "Probabilistic Machine Learning by David Barber".



Dean Abbott July 3, 2014 at 9:48 am #

REPLY ↩

Very nice taxonomy of methods. Two small quibbles, both in the Decision Tree section.

- 1) MARS isn't a tree method, it's a spline method. You list it already in the regression group, though could even go in the regularization group. (not a natural fit in any, IMHO).
- 2) Random Forests is an ensemble method and sticks out a bit in the trees group. Yes, they are trees, but so is the MART (TreeNet) and some flavors of Adaboost. Since you already have an ensembles and RF is already there, I think you can safely remove it from the Trees.

Again, you've done a great job with this list. Congrats!

Dean



Jason Brownlee August 22, 2015 at 4:49 pm #

REPLY ↩

Thanks Dean, I'll take your comments on board.



sravan August 6, 2014 at 8:41 pm #

REPLY ↩

Greate article. my knowledge in Machne learning is improving in bredth not in depth.how should i improve my learning.i have done some real time implementations with Regression analysis.and Random forest.and also i am atteding coursera courses.how would i get real time experience on ML R with Hadoop.



lale November 23, 2014 at 9:16 pm #

REPLY ↩

Thanks Mr.Brownly for your useful guide.Where can we find the implementations of all of these algorithms?I've installed weka but it doesnot have some of these algorithms



Jason Brownlee November 24, 2014 at 5:50 am #

REPLY ↩

You may have to make use of other platforms like R and scikit-learn.

Were you looking for an implementation of a specific algorithm?



SHI XUDONG November 25, 2014 at 2:42 pm #

REPLY ↩

Great Post!

I am currently learning Sparse Coding. And I have difficulty putting Sparse Coding into the categories you created.

–What is your idea about Sparse Coding?

–Which category should it belong to?

Can you provide some suggestions for learning sparse coding

– what mathematical foundations should I have?

– any good tutorial resources?

– can you suggest a learning roadmap

I am now taking convex optimization course. Is it a correct roadmap?



Lee January 13, 2015 at 8:48 pm #

REPLY ↩

Where does ranking fit into the machine learning algorithms? Is it by any chance under some of the categories mentioned in the article? The only time I find ranking mentioned in relation to machine learn is when I specifically search for ranking, none of the machine learning articles discuss it.



Amelie February 3, 2015 at 10:41 am #

REPLY ↩

which algorithm is the more efficient of the similarity algorithm .?



Jason Brownlee February 19, 2015 at 8:42 am #

REPLY ↩

Assess similarity algorithms using computational complexity and empirically test them and see Amelie.



Gudi February 13, 2015 at 3:31 pm #

REPLY ↩

What methods/algorithms are suitable for applying to trading patterns analysis. I mean looking at the trading graphs of the last 6 months (e.g. SPY). Currently, I am looking at the graphs visually. Can an algorithm come to my aid (I am currently enrolled in an online data mining course) ?



Jason Brownlee February 19, 2015 at 8:42 am #

REPLY ↩

Sounds like a timeseries problem, consider stating out with an auto-regression.



saima May 25, 2015 at 4:23 pm #

REPLY ↩

Hi Jason,
Its a great article. I wish if you could give a list of machine learning algorithms popular in medical research domain.

regards,
Saima Safdar



Vicc May 27, 2015 at 7:22 pm #

REPLY ↩

Great list. Definitely cleared things up for me, Jason! I do have a question concerning Batch Gradient Descent and the Normal Equation. Are these considered Estimators?

I would love to see a post that addresses the different types of estimators / optimizers that could be used for each of these algorithms that is simple to understand. Also where does feature scaling (min max scaling & standardization) and other things fall into all of this? Are they also optimizers? So many things!

Thanks so much for spreading your knowledge!



Henry Thornton June 6, 2015 at 10:49 pm #

REPLY ↩

Hi Jason

Intrigued by your comments above about recommendation systems ie.

"I would call recommender a higher-order system that internally is solving regression or classification problems." and,

"You can break a recommender down into a classification or a regression problem."

Could you please expand on your thought process? In general, I find that people talk about building or wanting a "classifier" since it is the de-jure buzzword (and related to deep learning) when in fact, a recommender or something else will do the job. Anyway, great discussion.



Aharon Robinson June 11, 2015 at 8:53 am #

REPLY ↩

Great stuff here Jason! Regarding your comments on 12/26, I'll vote yes to seeing a post on reinforcement learning methods



Vijay Lingesh June 11, 2015 at 4:13 pm #

REPLY ↩

Hi Jason,

I'm trying to implement object detection through computer vision through Machine Learning but I'm hitting a wall when trying to find a suitable approach. Can you suggest which kind of algorithm will help me? I'd like to research more on it.



Rajmohan July 16, 2015 at 3:44 pm #

REPLY ↩

Hi.. i am working on finding the missing values by using machine learning approaches..

Any body can suggest new methods to be used..
I am a research scholar



Oren August 5, 2015 at 7:04 pm #

REPLY ↩

Hi Jason,

just a small question: In my opinion k-NN, SVM, Naive Bayes, Decision Trees, MaxEnt (even if it's not mentioned here) are all considered to be instance-based, isn't it right?



Vaibhav Agarwal September 10, 2015 at 3:20 am #

REPLY ↩

Awesome post now I know where I stand.



shani September 10, 2015 at 10:07 pm #

REPLY ↩

i started reading and i feel i don't succeed to understand it.
I don't understand which algorithm is good for which type of problem.
I think that little example for each algorithm will be useful.



Gian September 22, 2015 at 11:30 pm #

REPLY ↩

Hi,

How can I classify the support vector machines and its extensions in your list?



Stephen Thompson October 7, 2015 at 1:25 am #

REPLY ↩

Jason: Nice addition of the simple graphic to each of the "families" of machine learning algorithms. This is a change from what I recall was a previous version of this post. The diagram helps visualize the activity of the family and thus aid developing an internal model of how the members of the family operate.

A simple but powerful effect.



Kevin Keane October 28, 2015 at 5:45 am #

REPLY ↩

The Bayesian Algorithms graphic should be reworked. In particular,

- 1) the area under both density functions should integrate to one. While no scale is provided, the prior appears to integrate to a much smaller number than the posterior.
- 2) in general, a posterior is narrower / more concentrated than a prior given an observation.
- 3) (interpreting the baseline as zero density) a posterior typically concentrates the probability of the prior in a smaller range; it never “moves” probability to a range where the prior density was zero.

Leave a Reply

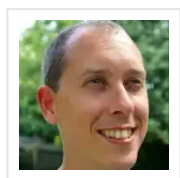
Name (required)

Email (will not be published) (required)

Website

SUBMIT COMMENT

Welcome!



Hi, my name is Jason and welcome to Machine Learning Mastery where we make programmers awesome at machine learning.

If you have any questions or you just want to chat about machine learning, [send me an email](#).

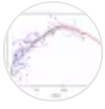
Free Guide to the Best Resources

Discover the best videos, books, courses and MUCH MORE in my Machine Learning Resource Guide.

Get your copy right now!

[Download For Free](#)

POPULAR



A Tour of Machine Learning Algorithms

NOVEMBER 25, 2013



How to Run Your First Classifier in Weka

FEBRUARY 17, 2014



Tutorial To Implement k-Nearest Neighbors in Python From Scratch

SEPTEMBER 12, 2014



How To Implement Naive Bayes From Scratch in Python

DECEMBER 8, 2014



Machine Learning for Programmers: Leap from developer to machine learning practitioner

AUGUST 17, 2015



Discover Feature Engineering, How to Engineer Features and How to Get Good at It

SEPTEMBER 26, 2014



Best Machine Learning Resources for Getting Started

NOVEMBER 27, 2013



Feature Selection with the Caret R Package

SEPTEMBER 22, 2014



Find Your Machine Learning Tribe: Get Started And Avoid Getting The Wrong Advice

AUGUST 31, 2015



4 Self-Study Machine Learning Projects

JANUARY 3, 2014

Looking for something? Search!



© 2015 Machine Learning Mastery. All Rights Reserved.

[Privacy](#) | [Contact](#) | [About](#)

