1 Introduction and Overview

1.5 Basic machine learning idea

We would like to take a sample of observations with associated class labels and try to learn (without explicitly defining it yet) so that when a new observation arrives we will process it and produce a class label that matches it.

If we would have known the real label we would have made practically no mistakes. The goal is to minimize mistakes made on newly arrived data.

The basic idea of machine learning can be demonstrated through inductive reasoning, which refers to the process of deriving conclusions from given facts. For example: if you have observed that your friend Mary has a Toyota, your uncle John has a Toyota, and your colleague at work also drives a Toyota, you may tend to think that everybody drives a Toyota. This is a simple but typical example of inductive reasoning; you have learned that “everybody drives a Toyota.” Of course induction is not sound, because it does not guarantee the correctness of the conclusion.

An important form of performing inductive reasoning is through generalization. Since induction is not sound, it is important to decide how good a conclusion is. The theory of learnability has been established for quantifying the effectiveness of learning. Learnability can be measured in terms of probability such that a class of concepts is considered learnable if an algorithm exists that executes efficiently and has a high probability of finding an approximately correct concept. (The theory based on this definition, probably approximately correct (PAC) will be discussed later)

Roughly speaking, learning refers to positive changes towards improved performance.

1.6 Different Learning Approaches

Store observations in memory and retrieve: An approach, also known as instance-based or case-based learning, which represents knowledge in terms of specific cases or experiences and relies on flexible matching methods to retrieve these cases and apply them to new situations. When a new observation arrives we hope that it has already been seen and remembered, we will scan the memory and output the same class label.

Even though we can employ enormous amounts of memory it will never be sufficient to guarantee good results. The dimensionality of the problems is much too big and we cannot rely on reoccurrences. This method was useful as long as the size of the datasets was small. (See Section 1.21 on growth of dimensionality)
To overcome this basic problem even if an observation of a certain type was not previously seen we would look for the “closest” past event or events and use their class label. This immediately requires a distance measure – the standard Euclidian one doesn’t always fit. A main question is how do you find a good distance measure?

The typical case-based learning method simply stores training instances in memory; generalization occurs at retrieval time, with the power residing in the indexing scheme, the similarity metric used to identify relevant cases, and the method for adapting cases to new situations.

**Learn a set of rules and apply to new data**: Relies on user defined rules, rather simple software which is based on a content specialist to feed rules into it. A “smart model” (relevant and full), as smart as the specialist, would result in a good generalization so that labeling of unobserved data will be done with a minimal error.

An example of such a system is a medical decision support system that Microsoft ([http://research.microsoft.com/research/dtg/](http://research.microsoft.com/research/dtg/)) tried to develop without much success. The main problem was that the system must always adapt and learn and not stay constant in order to deal with such complicated problems. This flourished the research on machine learning area.

More on this see 1.8

**Estimate a “flexible model” from the data**: Here “flexibility” means a method’s capability to estimate arbitrary dependencies from finite data. Parametric methods impose very stringent assumptions and are likely to fail if the true parametric form of a dependency is not known. On the other hand, classical non parametric methods do not depend on parametric assumptions, but they generally fail for high-dimensional problems with finite samples. Adaptive methods use flexible (very wide) class of approximation functions that can, in principle, approximate any continuous function with a pre-specified accuracy. This is known as Universal approximation property.

*Parametric methods* use a very restricted set of approximating functions of pre-specified parametric form, so only a fixed number of parameters need to be determined from the data.

*Flexible methods* use a wide set of functions (universal approximators) capable of approximating any continuous mapping. The class of approximating functions used by flexible methods is thus very wide and allows for multiple solutions when a model is estimated with finite data.

1.8 Learning a Set of Rules

This is a hybrid between machine learning and user defined set of rules which is further easily interpreted and thus provides a good feedback to the user about the model.

**Tree structures**: A simple model that defines an expert system in an easy way. A combination between the ability to learn the model (and interpolate) and conclusions of what it reveals about the nature of the data. The accuracy goal to reach an error of less than 5% is not always the most important factor. For example in a financial prediction 51% hit rate is enough to make millions of dollars, you don’t need 95%, but the problems are very complex. In medicine it is less important to predict a patients survival chances in high accuracy but rather
to point out the major factor that affects his survival chances. Thus the interpretation is much more important than the result.

**Graphical models:** Graphical models are a merge between probability theory and graph theory. They provide a natural tool for dealing with two problems that occur throughout applied mathematics and engineering -- uncertainty and complexity -- and in particular they are playing an increasingly important role in the design and analysis of machine learning algorithms. Fundamental to the idea of a graphical model is the notion of modularity -- a complex system is built by combining simpler parts. Probability theory provides the glue whereby the parts are combined, ensuring that the system as a whole is consistent, and providing ways to interface models to data. The graph theoretic side of graphical models provides both an intuitively appealing interface by which humans can model highly-interacting sets of variables as well as a data structure that lends itself naturally to the design of efficient general-purpose algorithms.

[http://www.ai.mit.edu/~murphyk/Bayes/bnintro.html#repr](http://www.ai.mit.edu/~murphyk/Bayes/bnintro.html#repr)

When you derive a set of rules from data it is interesting to measure the complexity of the set with its ability to generalize. For example when you need to decide if a patients problem stems from the heart or liver the question is if you need to drill down through too many details to find out or would a small set of tests be enough?

### 1.10 Applications

#### Control

Robot arm: A few years ago there was an assumption that this issue is solved but it is not valid any more. Due to the 6 freedom levels it turns out that the problem is extremely non linear. Not every observation has a label and only a series of observations has so the learning is much more complicated (reinforcement learning [http://www-anw.cs.umass.edu/~rich/book/the-book.html](http://www-anw.cs.umass.edu/~rich/book/the-book.html))

Driving and navigating a car: Mercedes-Benz are working on the issue and an Israeli company called Mobileye are in advanced stages of a single camera driving assistance solution [http://www.mobileye.com/](http://www.mobileye.com/)

Medical applications: The FDA is holding back on initiatives to replace the physicians in the diagnosis phase (mainly due to liability issues) but in the monitoring and drug release areas we have seen some progress in the last few years.

#### Web Retrieval based on user profile

Amazon: are doing a deep analysis of the way people buy and how to label their areas of interest so that they can offer them automatically exactly the things they would like to buy. [http://www.informatics.indiana.edu/fil/Papers/customshopper.pdf](http://www.informatics.indiana.edu/fil/Papers/customshopper.pdf)

Google: is one of the leading search engines in the internet and has peaked the top for a long time with only a few other survivors. [http://www.google.com/technology/index.html](http://www.google.com/technology/index.html)
1.11 Related Disciplines

- AI – was the former name for machine learning which didn’t solve problems methodically but by a set of heuristics.

- Decision Theory – related to Game Theory.

- Control Theory – used in robot arm control. It is amazing to see how people sit for days to design complicated flow models for specific instances of problems and how Control Theory gives in many cases more robust, agile and outperforming results. [http://www.math.rutgers.edu/~sontag/FTP_DIR/mct-intro.pdf](http://www.math.rutgers.edu/~sontag/FTP_DIR/mct-intro.pdf)

- Information Theory – Most of the principles we will learn about stem from this widest theoretical basis.

- Philosophy - How, What and When we learn.

- Psychology – Compares the accomplishments of people or animals and infers how machines should learn. To decipher how the brain works. The brain as a machine is utilizing advanced signal processing.

A bat is performing rapid signal processing and 3D analysis of the space around it from simple sonar and two omni-directional ears. The evolution created amazing results: [http://www.cs.tau.ac.il/~nin/papers/TimeFrequencyCompact.pdf](http://www.cs.tau.ac.il/~nin/papers/TimeFrequencyCompact.pdf)


- Computational Complexity Theory – used to analyze algorithms and compare theoretically which of them is better.

- Data Mining - Data Mining is the automated extraction of hidden predictive information from databases.

Data mining software allows users to analyze large databases to solve business decision problems. Data mining is, in some ways, an extension of statistics, with a few artificial intelligence and machine learning twists thrown in. Like statistics, data mining is not a business solution, it is just a technology. For example, consider a catalog retailer who needs to decide who should receive information about a new product. The information operated on by the data mining process is contained in a historical database of previous interactions with customers and the features associated with them, such as age, zip code, and their responses. The data mining software would use this historical information to build a model of customer behavior that could be used to predict which customers would be likely to respond to the new product. By using this information a marketing manager can select only the customers who are most likely to respond. The operational business software can then feed the results of the decision to the appropriate touch point systems (call centers, direct mail, web servers, email systems, etc.) so that the right customers receive the right offers.
1.12-13 Credit Risk Analysis

Consider a bank that would like to grant credit only to "good" customers. The bank has a lot of information about its customers, such as age, income, marital status, financial reputation, etc. When a customer requests a line of credit, the bank must decide whether or not to approve such a request. Traditionally, the criteria for granting or denying credit are set by human beings.

A system based on rules that were derived from a small set of observations did not function well enough. If someone would have written a system like this it would have been fairly easy to con and get a loan even if you don’t intend to return it. The system must change and adapt to the behavior of the users and to the changing macro financial conditions. A static system can never be efficient enough.

1.14 Clustering news

In analyzing text the class label isn’t really given. When we look at people’s behavior when surfing the web and try to extrapolate what will be the next URLs of interest there are many class labels. One person would go to the sports column and the other to the science one. Methods that are based on people’s profile like cookies would yield better results (the ads would be more relevant and the sales would rise). This is the main method in which commercials will be done in the next few years.

TiVo (http://www.tivo.com/5.3.1.1.asp?article=200) turns out to be a much worse “nightmare” than what Orwell had anticipated. When you buy TiVo you are in effect being monitored by the company which gathers all your TV habits: which commercials you watch, which shows, how many times you use the replay feature and on what sections etc.

The idea is that expert systems would customize the content to the customer for perfect fit.

1.16 Medical applications

This is a relatively new field which is growing very rapidly. It is aimed at helping doctors make decisions, and helping patients monitoring in and out of a hospital. A novel field of applications is patient’s monitoring for the purpose of automatic drug delivery.

One of the problems in medical data is that it is not so easy to get to a diagnosis. Even with human doctors a 2nd opinion is highly recommended. It stands out when you see doctors explore X-Ray photos – even though they hold the highest salaries their prediction rate is still way under 50%.

Even that the imaging technologies have gone through a lot of advancement both in resolution and noise reduction the ability to classify a tumor as a benign or cancer type are still weak.

Medical imaging data compression is of high interest due to the enormous amount of digital data produced even in a small hospital and the long years it has to be kept. A lot of companies tried to find a suitable compression which does not decrease the physicians hit rate.
In medical applications the cost of error is not symmetric. If a doctor decides to send someone to a futile operation it's the cost of the operation but if he wrongfully doesn't it could be the cost of the patients' life.

1.18-21 History of machine learning

The field is relatively new and has changed a lot in the last 10-15 years. In the 60’s (Minsky at MIT) there was a belief that the problem of Artificial Intelligence is on the verge of solution. BBC – “In a months time the vision problem is solved” but today we still can't build a robot smart enough to outperform a 3 year old child in navigating in a room with obstacles.

Genetic Algorithms: The evolution took so many years that it refined the representation and processing to their current enviable levels. Genetic algorithms try to imitate this process in a computational way.

(1.19) During the years some concepts have emerged and vanished, but have kept a lot of researchers busy, like learning from instructions, by analogy and from cases.

(1.20) In the early 80s a team, based out of UCSD, headed by Jeff Hinton published a huge amount of new findings amongst them was the back propagation which revived the neural network paradigm.

(1.21) The Perceptron, offered by Minsky & Papert, showed that you can solve learning problems without extensively writing the rules. [http://home.cc.umanitoba.ca/~umcorbe9/perceptron.html](http://home.cc.umanitoba.ca/~umcorbe9/perceptron.html)

The **Curse of dimensionality** (Bellman 1961) refers to the exponential growth of hyper-volume as a function of dimensionality. A priori information can help with the curse of dimensionality. Careful feature selection and scaling of the inputs fundamentally affects the severity of the problem, we will learn ways to deal with this fundamental problem.

From 1980 to 1986 a group in Stanford worked on flexible models. They understood that there is no point in trying to tackle high dimensionality problems with little data.

The theory is in lockstep behind the business demand and the new types and dimensions of data collected today – we do not really know how to solve the problems we deal with daily.

1.23 Types of models

Learning usually involves a set of samples $S$ (The ‘Training Set’ or ‘Training Data’). Each sample $s \in S$ has a label (classification). We can think of learning as a process by which we try to formulate a classification function that takes as input some $s \in S$ and, with high probability, outputs its correct label. Furthermore, we would like this function to be able to output the correct label, with high probability, to any $s$ in our universe of samples, that is, even for $s \notin S$.

There is more than one model for machine learning, A few are:

**Supervised learning**: The algorithm is given access to a classified data train set $S$ (a label is provided for each sample in $S$) and will try to formulate classification function accordingly. In this kind of learning we have to be especially careful of
over-fitting, that is, the algorithm will classify samples from $S$ very well but will not be successful in classifying samples that are not in $S$ (or in other words it would fail to generalize, and will classify $S$ according to some irrelevant noise, more on this later).

**Unsupervised learning:** The algorithm is given access to data, but no classification is given. That is, the label of each sample is not given and the algorithm has to find some ‘pattern’ in the samples and use it for formulating its classification function. One of the dangers of this method is getting unwanted results: the algorithm can classify the data in some unexpected manner. This model is important for data reduction: more is learned out of fewer samples.

**Control learning:** Selects actions and observes consequences (Maximizes long-term cumulative return.) This model is used mainly for mechanical learning problems such as moving an arm of a robot. It is used where there is a difficulty to separate different actions and obtain individual feedbacks (labels) for each.

---

**Exempli Gratia**

Assume we want to train an android (a synthetic human) to be a Taxi Driver

**Supervised Learning**

We put the android in the driver seat of the Taxi and place an instructor next to it. Every time there is a need to brake, the instructor shouts “brake!”. The android can learn from each driving condition (sample) and instructor response (label) when to brake.

**Unsupervised Learning**

Overtime of working as a cabby we would expect our android to develop an understanding of where and when there is likely to be heavy traffic and learn to avoid these areas without being instructed.

**Control Learning**

The android may or may not receive tips from passengers. By selecting different actions (talking about politics with passengers, driving really fast etc.) and observing the amount of tips he receives, he can modify his performance to maximize his cumulative return (amount of tips).

---

### 1.24-25 Complete information

Let us look into a simple example of learning. We are given a two dimensional space (a square) containing a data set that is classified into two groups: pluses (+) and minuses (-) as shown in figure 1 on the right.

There is complete information, that is, the distribution functions of the pluses and minuses are known; let us denote them D+ and D-. We allow ourselves another simplification: we assume the pluses and minuses are equally likely (i.e. $P(+) = P(-) = \frac{1}{2}$).

We want to classify a new point $(x,y)$, i.e. to determine whether it is a plus or a minus. To achieve this we will calculate the probability of receiving a plus at $(x,y)$ using Bayes’ formula:

![Figure 1 Classified data set](image-url)
We will classify \((x,y)\) as a plus if \(p>0.5\), otherwise as a minus. Note that our errors will be divided symmetrically: the number of pluses classified as minuses and the number of minuses classified as pluses will be similar (their expected values are equal).

This approach is not always the best. Different errors may have different costs. Consider for example an intelligence bureau that is trying to determine whether a terrorist attack is about to take place.

On one hand, wrongfully predicting a terrorist attack will occur has a comparatively negligible cost in terms of heightened security measures. On the other hand failing to predict a terrorist attack that will take place will have dire consequences in terms of loss of human lives and damages.

This brings us to the subject of loss models.

### 1.26 Predictions and loss models

Let \(p\) be some fact which we are trying to predict (e.g. \(p=P(+(x,y))\) in the plus-minus example in 2.2). The general idea is to define some loss function \(L\) for wrong predictions and try to minimize it.

**Boolean error:** Is the rudimentary loss model and loss is defined as 1 for incorrect predictions and obviously a loss of 0 for correct predictions. Since \(p=P(+(x,y))\) and \(1-p=P(-|(x,y))\) we predict ‘+’ if \(p>0.5\), otherwise we predict ‘-’.

Note that in this model \(p\) cannot be recovered from the results: \(p=0.51\) and \(p=0.99\) will give the same prediction.

**Quadratic loss:** Is an example of a more complex model in which we choose a “real number” \(q\), where \(p\) is defined as before, and define the loss to be:

\[
(p-q)^2 \quad \text{For incorrect plus signs}
\]

\[
([1-p]-[1-q])^2 \quad \text{For incorrect minus signs.}
\]

Note that:

\[
([1-p]-[1-q])^2 = (q-p)^2
\]

The expected loss \(L = (q-p)^2\) (because \(L = p(p-q)^2 + (1-p)(q-p)^2\)). The minimal loss is obviously achieved when \(p=q\), but note that \(p\) might be unknown. The knowledge of \(p\) is needed in order to calculate the loss but this model recovers the probabilities: different \(p\) yield different losses.

### 1.27 Bayes Estimator

**Bayes Estimator:** A Bayes estimator \(d\) associated with a prior distribution \(p\) and a loss function \(L\) is an estimator function that attempts to minimize the loss. For
example we can think of the quadratic mode in 2.3.2 as a Bayesian Estimator (2.3.2) with $L$ defined as the quadratic loss and where $q$ is the estimator - it attempts to estimate $p$.

**Bayes Risk Function:** A Bayes risk function $R$ is the expected value of the loss for a given estimator $d$. Note that the function $R$ is defined after choosing a decision procedure (in the quadratic loss example above we chose $q$ as an estimator). Formally: $R(p) = E[L(p)]$

The risk function gives us the ‘price’ we pay for using a specific decision procedure. In the plus-minus example it takes into account all possible plus and minus distributions.

### 1.29-30 The basic PAC model

The PAC (Probably Approximately Correct) model sets out to formalize the learning process. It is a batch-learning model i.e. the algorithm is trained over some fixed data set. We assume that the distribution $D$ is fixed, although unknown, over the sample space $X$. Our target function is $f$ (for every $x \in X$, $f$ yields the correct prediction). Given a hypothesis $h$ we define its error:

$$e(h) = P_{x \sim D} [h(x) \neq f(x)]$$

Our goal is to find a hypothesis $h$ that minimizes the error. In practice $h$ must be chosen out of a limited family of hypothesis in order to simplify the search. One common approach is to choose a small family $H$ and extend it in order to make the error smaller.

Let’s examine a simple example. The sample space $X$ will be the segment $[0,1]$. The target function will be $f(x) = x^3$.

- We start out by choosing $H$ to be all linear functions ($h(x) = ax + c$). We find the best hypothesis $h1$, which is a line which approximates $f$. Note that the error is positive ($e(h1) > 0$).
- We extend our search to all polynomial functions of the second degree ($h(x) = ax^2 + bx + c$). In this case the best hypothesis $h2$ will yield a smaller error: $e(h1) > e(h2) > 0$ (it is still positive because $f$ is of the third degree)
- At this point we decide that we are satisfied with the error and stop the search. But there is a problem with this model: $D$ was used to calculate the error. But since the distribution $D$ is unknown it cannot be used. Instead, we are provided with a training data set of $m$ samples denoted as $S$. We define an estimation of the error:

$$e'(h) = \frac{|\{x \in S : h(x) \neq f(x)\}|}{m}$$

(Outputs for $h$ the fraction of samples that it classified incorrectly)

The basic question we now need to attend to is how close are $e'(h)$ and $e(h)$.

### 1.31 Bayesian Theory

$S$ is a set of observed samples and is used as training data. $h$ is some hypothesis; $h \in H$, where $H$ is the set of possible hypothesis.

**Bayes theorem**
Prior Probability: \( P(h) \) - The initial probability that hypothesis \( h \) holds, before we observed training data \( S \). Often called the prior probability of \( h \) and may reflect background knowledge we have about the chance \( h \) to be the correct hypothesis. Usually, if we have no such prior knowledge we will assign an equal probability for each candidate hypothesis \( h \).

Maximum Likelihood (ML): \( P(S \mid h) \) - The probability of observing data \( S \), given a world where hypothesis \( h \) holds. Finding an \( h \) that maximizes this term will give us a hypothesis that maximizes the likelihood of observing \( S \) in our world.

Maximum A Posteriori (MAP): \( P(h \mid S) \) - called the posterior probability of \( h \), because it reflects our confidence that \( h \) holds after (hence the word posterior) we have seen training data \( S \). In many cases we are interested in finding hypothesis \( h \) with the maximum a posteriori value.

Bayesian Predictor: \( \sum_{h \in S} h(x)P(h \mid S) \) - Sums all hypotheses each multiplied by its posterior probability. Analogous to getting advice from a few experts, for each we have a different confidence level, and summing it up.

1.32-33 Some issues in Machine Learning

We will try to understand how and when prior knowledge helps. Also we are interested in the effect the number of training samples we have on our success.

1.34 Complexity vs. Generalization

We would like to compare the complexity of our hypothesis vs. the observed error. A more complex hypothesis has better results for a given training set. So it seems natural to think that enlarging the family of hypothesis \( H \) should only improve our results. But in reality sometimes simple models have better results because they ignore noise in the data. Consider the following data set:

Below is a simple way of classifying the data, which allows errors on the samples but has the advantage of simplicity:

Following the lines of the concept of ‘Okham’s razor’ (or ‘Occam’s razor’) we will define measures to estimate the simplicity of a hypothesis. As usual we will attempt to minimize these measures. One example is Minimum Description Length (MDL):

\[
e'(h) + |\text{code length of } h|
\]
Another measure is *Structural Risk Minimization* (SRM):

\[ e'(h) + \left( \log |H| / m \right)^{0.5} \]

Note that the SRM model is dependent on the size of the whole family \( H \) (we penalize for using a large family of hypothesis).

### 1.36 Weak Learning

We assume that for any distribution \( D \) there is some weak predicate \( p \), which predicts better than \( \frac{1}{2} + \epsilon \) (is at least slightly better than tossing a coin to determine classification). Using a *Boosting Algorithm* on this predicate we will improve it in order to reach strong learning:

\[ \text{Multiple weak learning} \xrightarrow{\text{Boosting}} \text{strong learning} \]

Boosting algorithms are efficient and have shown good experimental results. They are currently very popular with researchers of Machine Learning.