1 Introduction

We have been talking coding theory and information theory in the first three weeks of this course. And error correcting codes (tornado codes and erasure channels) in the previous two weeks. We will focus on classification, clustering and learning for this week.

Materials of this lecture can be found in Chapter 6 of the book Machine Learning by Tom Mitchell.

Machine learning has interesting connection to the information theory. However, it is a broad topic in itself. We cannot do justice to it in a week. But will offer a flavor of those popular techniques currently in the field.

Large amount of data exist in different applications. For example, web page collections, text documents, credit histories and symptoms for patients. One may want to cluster or classify these documents to obtain useful information such as the category of a given web page. Many learning algorithms can cluster or classify. We will mainly deal with the baysian approach, which assumes prior information. The baysian approach has simple and sound mathematics foundation. However it needs a fair amount of prior information and needs computation for every hypothesis in model parameter space to obtain the optimum model.

2 Bayes Theorem, MAP and ML

Theorem 2.1. *H hypothesis*
D data
$P(h)$ probability of a particular hypothesis h
$P(D|h)$ probability of observing D given h

$P(h|D) = \frac{P(D|h) \cdot P(h)}{P(D)}$

1. MAP (Maximum a posterior hypothesis that maximize $P(h|D)$)
   $h_{MAP} = \arg\max_{h \in H} P(D|h) \cdot P(h)$
2. ML (Maximum likelihood hypothesis)
   $h_{ML} = \arg\max_{h \in H} P(D|h)$

If priors are uniform, $h_{ML} = h_{MAP}$.

Example: suppose we are conducting a test to see if a patient has a certain kind of cancer or not. Assume prior of getting this cancer for a population is 0.8%. If a patient has cancer, probability of positive test results is

Solution:

1. if hypothesis is having cancer, $P(positive|cancer) \cdot P(cancer) = 0.98 \cdot 0.008 = 0.00784$
2. if hypothesis is having no cancer, $P(positive|!cancer) \cdot P(notcancer) = 0.03 \cdot 0.992 = 0.02976$

Therefore the MAP hypothesis if given a positive test result is no cancer at all.

3 Concept Learning

We define a hypothesis space H and a mapping function $c : x \rightarrow \{0, 1\}$. The mapping function defines the concept of a given data point. This mapping is the focus of what a learner wants to learn. We formulate the concept learning problem as follows:

Given a set of values (x_i, d_i), where $d_i = c(x_i)$. Find $h \in H$, such that $d_i = h(x_i)$.

We assume

1. training data is noise free
2. target concept $c \in H$
3. no *apriori* reason to favor $h_1 \in H$ or $h_2 \in H$

If the concept found by a hypothesis agrees with the given label of a data point for all the data, we say the hypothesis is consistent with given data. Express more theorectially,

$$P(D|h) = \begin{cases} 1 & \text{if } d_i = h(x_i) \forall d_i; \\ 0 & \text{otherwise.} \end{cases}$$

The hypothesis is consistent with the given data if and only if $P(D|h) = 1$.

We define **version space** as

$$\begin{aligned} VS_{H,D} &= \text{version space of } H, D \\ &= \{h \in H, h \text{ consistent with } D\} \end{aligned}$$

Any single h in $VS_{H,D}$ is h_{MAP}.

$P(h|D) = \frac{1}{|VS_{H,D}|}$

Now we relax one assumption when defining concept learning. We will assume the data is noisy instead of noisefree from now on. Reformulate the problem as

we want to learn some function $f : X \rightarrow \Re$ such that given $(x_i, d_i), d_i = f(x_i) + e_i$ if errors (e_i) are normally distributed, how to choose f?

$$\begin{aligned} h_{MAP} &= \arg\max_{h \in H} P(D|h) \\ &= \arg\max_{h \in H} \prod_i \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{1}{2} \cdot (\left(\frac{d_i - h(x_i)}{\sigma}\right))^2} \end{aligned}$$

Take logs, assume σ is fixed

$h_{ML} = \arg\min_{h \in H} \sum_i (d_i - h(x_i))^2$

Why we are using Gaussians to model the errors of learning functions? First of all Gaussians have nice properties. Secondly it does approximate the shape of the error distribution curve.

How do we choose a probability distribution using MAP or ML?

Maximum likelihood hypothesis for probabilities: $f : X \rightarrow \{0, 1\}$, a probabilistic mapping function $f'(x) = prob[f(x) = 1]$

given a function space (binary valued functions in this case), what is the best function to choose?

We want to maximize $P(D|h)$ where $D = \{\langle x_i, d_i \rangle\}$

Assume samples independent of each other,

$$
\begin{aligned}
P(D|h) &= \prod_{i=1}^{n} P(x_i, d_i|h) \\
&= \prod_{i=1}^{n} P(d_i|h, x_i) \cdot P(x_i)
\end{aligned}
$$

$$
\begin{aligned}
P(d_i|h, x_i) &= \begin{cases} h(x_i) & \text{if } d_i = 1; \\ 1 - h(x_i) & \text{if } d_i = 0. \end{cases} \\
&= (h(x_i))^{d_i} \cdot (1 - h(x_i))^{1-d_i}
\end{aligned}
$$

$$
\begin{aligned}
h_{ML} &= \arg\max_{h \in H} \prod_i P(d_i|h, x_i) \\
&= \arg\max_{h \in H} \prod_i h(x_i)^{d_i} \cdot (1 - h(x_i))^{1-d_i}
\end{aligned}
$$

Take logs,

$$
h_{ML} = \arg\min_{h \in H} \sum_i d_i \cdot \log h(x_i)(1 - d_i) \cdot log\,(1 - h(x_i))
$$

The constituent terms of the sum can be interpreted as cross entropy in information theory. Cross entropy measure distance between observed distribution and correct distribution. When the mapping functions isnt probabilistic, we are using squared distance instead of cross entropy.

4 MDL (Maximum Description Length)

MDL principle: of all the hypothesis you can choose from, choose the one with minimum description length.

We will explain this principle by MAP hypothesis.

$$\begin{aligned} h_{MAP} &= \arg\max_{h\in H} P(D|h)\cdot P(h) \\ &= \arg\max_{h\in H} log_2 P(D|h) + log_2 P(h) \\ &= \arg\min_{h\in H} log_2 P(D|h) - log_2 P(h) \end{aligned}$$

$-log_2 P(h)$: size of optimum encoding of h. Each h in hypothesis space will associate with a probability according to prior information. Therefore there is a probabilistic distribution of hypothesis in the space. The optimum encoding of this distribution gives the size of optimum encoding of a particular h.

$-log_2 P(D|h)$: size of the optimal encoding of the data given h.

Therefore h_{MAP} picks the hypothesis which minimize the size of optimal encoding of the model (hypothesis) and optimal encoding of the data given the model. Given c_1, c_2 being two encodings which encode respectively the model and the data given model,

$h_{MAP} = \arg\min_{h\in H} L_{c_1}(h) + L_{c_2}(D|h)$

Baysian approach justifies the MDL principle by saying whenever to choose models, choose the simpler one. It is always possible to have a complicated model tuned extremely well for training data. However such models are not likely to generalize to unseen data. By choosing the simpler model, the problem of overfitting can be alleviated.

5 Bayes Optimal Rule

Example:

h_1, h_2, h_3 are three hypotheses. Given a sample x and its classification result under the hypostheses,

h_1	+	0.4
h_2	-	0.3
h_3	-	0.3

What is the most likely classification? h_{MAP}=positive
however if we weight each hypothesis decision by their probabilities,

$$\begin{aligned} p[+] &= 0.4 \\ p[-] &= 0.6 \end{aligned}$$

Therefore the most likely classification is negative.

Here we introduce the bayes optimal rule.

Choose the hypothesis which satisfies the following equation where $v_j \in V$ are labels:

$$\arg\max_{v_j\in V} \sum_{h_i\in H} P(v_j|h_i)\cdot P(h_i|D)$$

Given hypothesis space, given prior distribution, bayes optimal rul minimizes the classification errors of new samples.

However, even if we can learn hypothesis quickly, we still have to sum up over all hypothesis space. People have devised workaround algorithms to curtail this problem. Gibbs algorithm is one of them.

Gibbs Algorithm:

1. pick $h \in H$ according to *aposterior* distribution
2. classify new example using h

The expected misclassification error of Gibbs algorithm is smaller or equal to two times the optimal misclassification error. Proof will be left to homework.

6 Nave Baysian Classifier

Given a data set, each data item x has attributes $\langle a_1, \cdots, a_n \rangle$, we want to know $f(x) \in V$,

$$\begin{aligned} v_{MAP} &= \arg\max_{h \in H} P(v_j | a_1, \cdots, a_n) \\ &= \arg\max_{h \in H} \frac{P(a_1, \cdots, a_n) \cdot P(v_j)}{P(a_1, \cdots, a_n)} \\ &= \arg\max_{h \in H} P(a_1, \cdots, a_n) \cdot P(v_j) \end{aligned}$$

However $P(a_1, \cdots, a_n)$ is hard to estimate because too many training data are required. Nave baysian classifier suggests

$v_{MAP} = \arg\max_{h \in H} P(a_1|v_j) \cdots P(a_n|v_j)$

This usually serves as the benchmark for experimenting with other types of classifiers.

7 EM algorithm

In the previous discussion we assume all parameters of a model are observed. However there are cases where some parameters are observed and some are not. EM algorithm tackles this problem by trying to estimate all the parameters, both observed and hidden. We will examine EM algorithm in next lecture and especially treat a variant of it: how to learn a mixture of Gaussians.