5. Bayesian Learning

References
T. Mitchell. Machine Learning. Chapter 6

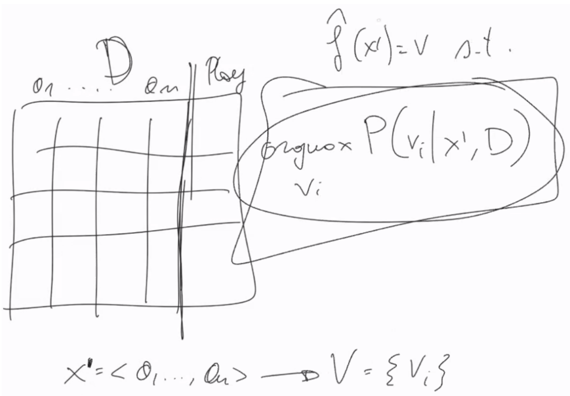
5.1 Bayes Methods

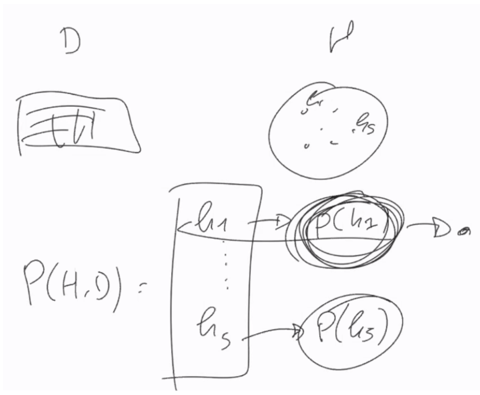
Provide **practical learning algorithms** and **conceptual frameworks**

Probabilistic estimation

Classification as Probabilistic estimation

Given f to learn : X → V, D (based on which f will learn), a new instance x', best prediction f(x') = v* ⇒ v* = argmax P(v|x',D) (v that maximizes prob)





Generally we want the most probable hypothesis h **given D**, hence, the ***Maximum a posteriori*** hypothesis h map: (we look for h that maximizes probability):

$$
\begin{aligned}
h_{MAP} &\equiv \arg\max_{h\in H} P(h|D) = \arg\max_{h\in H} \frac{P(D|h)P(h)}{P(D)} \\
&= \arg\max_{h\in H} P(D|h)P(h)
\end{aligned}
$$

Maximum Likelihood hypothesis of generating Data we are observing

$$h_{ML} = \arg\max_{h\in H} P(D|h)$$

N.B.: h MAP(x') may not be the most probable classification !!! (we take the class returned)

Bayes Optimal Classifier

Consider target function f : X → V, V = {v1, ..., vk}, data set D and a new instance x !in D:

P(vj |x,D) = SUM P(vj |x, hi)P(hi |D)

(given new example and D, the new example is classified as vj)

We are independent from the dataset, because, hi are given, so they are independent from other hypothesis based on dataset.

$$\underbrace{P(+|x,D)}_{} = (P(+|x,h_1,D) \cdot P(h_1|D)$$
$$+$$
$$P(+|x,h_2,D) \cdot P(h_2|D)$$
$$+$$
$$P(+|x,h_3,D) \cdot P(h_3|D)$$

Computes most prob class, Vob, for new instance x:

$$v_{OB} = \arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|x,h_i)P(h_i|D)$$

Example:

$$\begin{array}{lll} P(h_1|D) = 0.4, & P(\ominus|x,h_1) = 0, & P(\oplus|x,h_1) = 1 \\ P(h_2|D) = 0.3, & P(\ominus|x,h_2) = 1, & P(\oplus|x,h_2) = 0 \\ P(h_3|D) = 0.3, & P(\ominus|x,h_3) = 1, & P(\oplus|x,h_3) = 0 \end{array}$$

therefore

$$\begin{aligned} \sum_{h_i \in H} P(\oplus|x,h_i)P(h_i|D) &= 0.4 \\ \sum_{h_i \in H} P(\ominus|x,h_i)P(h_i|D) &= 0.6 \end{aligned}$$

and

$$v_{OB} = \arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|x,h_i)P(h_i|D) = \ominus$$

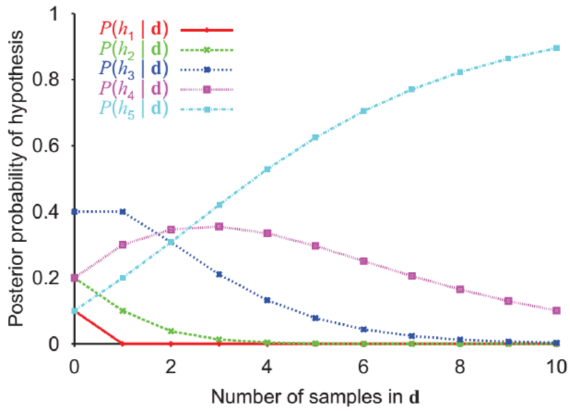
Optimal learner: no other classification method (same hyp space etc.) can outperform this one; it maximizes prob. new instance is classified correctly. Label new instances.

alfa in Bayes rule:

1. First candy is lime: $D_1 = \{l\}$

$P(h_i|\{d_1\}) = \alpha P(\{d_1\}|h_i)P(h_i)$ (Bayes rule)

$$\begin{aligned}
\mathrm{P}(H|D_1) &= \underbrace{\alpha}_{1/P(D_1)} \underbrace{< 0, 0.25, 0.5, 0.75, 1 >}_{P(D_1|H)} \cdot \underbrace{< 0.1, 0.2, 0.4, 0.2, 0.1 >}_{P(H)} \\
&= \alpha < 0, 0.05, 0.2, 0.15, 0.1 > \\
&= < 0, 0.1, 0.4, 0.3, 0.2 >
\end{aligned}$$



New example: consider theta = number of cherries in [0,1]

Data set: D = {c cherries, l lime}, N = c + l
P(c|h theta) = theta
P(l |h theta) = 1 – theta

$$h_{ML} = \underset{h_\theta}{\operatorname{argmax}} P(D|h_\theta) = \underset{h_\theta}{\operatorname{argmax}} L(D|h_\theta)$$

with $L(D|h_\theta) = \log P(D|h_\theta)$

$$P(D|h_\theta) = \prod_{j=1\ldots N} P(d_j|h_\theta) = \theta^c \cdot (1-\theta)^l$$

$$L(D|h_\theta) = c \log \theta + l \log(1-\theta)$$

$$\frac{dL(D|h_\theta)}{d\theta} = \frac{c}{\theta} - \frac{l}{1-\theta} = 0 \Rightarrow \theta_{ML} = \frac{c}{c+l} = \frac{c}{N}$$

$d_1 \; d_2 \; d_3 \; \ldots \; d_n$

$$\underbrace{D = \{l, l, l, c, c, l, c, l\}}$$

$$P(D|h_\theta) = \theta^3 \cdot (1-\theta)^5$$

Theta ml: is the proportion that best explains the data you are observing; it's the ratio of cherries over total in your bag and theta ML is the number that maximizes the probability of seeing the data you are observing given the hypothesis Htheta.
Theta ml is the most probable class you can obtain.

In general:

Theta is a vector of parameters.

$$\Theta_{ML} = \underset{\Theta}{\operatorname{argmax}} \log P(d_i|\Theta)$$

Bernoulli

Probability distribution of a binary random variable $X \in \{0, 1\}$

$$P(X = 1) = \theta \quad P(X = 0) = 1 - \theta$$

(e.g., observing head after flipping a coin, extracting a lime candy, ...).

$$P(X = k; \theta) = \theta^k (1 - \theta)^{1-k}$$

Multi-variate Bernoulli

Joint probability distribution of independent variables

$$P(X_1 = k_1, \dots ; \theta_1, \dots, \theta_n) = \prod_{i=1}^{n} P(X_i = k_i; \theta_i) = \prod_{i=1}^{n} \theta_i^{k_i}(1 - \theta_i)^{1-k_i}$$

Binomial

Probability distribution of k outcomes from n Bernoulli trials

$$P(X = k; n, \theta) = \binom{n}{k} \theta^k (1 - \theta)^{n-k}$$

Multinomial

Generalization of binomial

$$P(X_1 = k_1, \dots, X_d = k_d; n, \theta_1, \dots, \theta_d) = \frac{n!}{k_1! \dots k_n!} \theta_1^{k_1} \cdot \dots \cdot \theta_d^{k_d}$$

(e.g., rolling a d-sided dice n times and observing k times a particular value, extracting k lime candies after n extractions form a bag containing d different flavors, ...).

Summary

Probabilistic method: not eff cause you have to calculate all the distributions
Maximum likelihood: it's convenient because you can simply iterate calculus

5.2 Naive Bayes Classifier

Naive Bayes Classifier uses conditional independence to approximate the solution; works under the assumptions that are independent.

$P(X,Y \mid Z) = P(X \mid Y, Z)P(Y \mid Z) = P(X \mid Z)P(Y \mid Z)$

$$
\begin{aligned}
v_{MAP} &= \underset{v_j \in V}{\operatorname{argmax}} P(v_j | a_1, a_2 \ldots a_n, D) \\
&= \underset{v_j \in V}{\operatorname{argmax}} \frac{P(a_1, a_2 \ldots a_n | v_j, D) P(v_j | D)}{P(a_1, a_2 \ldots a_n | D)} \\
&= \underset{v_j \in V}{\operatorname{argmax}} P(a_1, a_2 \ldots a_n | v_j, D) P(v_j | D)
\end{aligned}
$$

Bayes rule

Eliminate denominator because is positive

Class of new instance x:

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j | D) \prod_i P(a_i | v_j, D)$$

5.2.1 Naive Bayes Algorithm

Target function $f : X \mapsto V$, $X = A_1 \times \ldots \times A_n$, $V = \{v_1, ..., v_k\}$, data set D, new instance $x = \langle a_1, a_2 \ldots a_n \rangle$.

Naive_Bayes_Learn(A, V, D)

for each target value $v_j \in V$

$\hat{P}(v_j | D) \leftarrow$ estimate $P(v_j | D)$

for each attribute A_k

for each attribute value $a_i \in A_k$

$\hat{P}(a_i | v_j, D) \leftarrow$ estimate $P(a_i | v_j, D)$

Classify_New_Instance(x)

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} \hat{P}(v_j | D) \prod_{a_i \in x} \hat{P}(a_i | v_j, D)$$

$$\hat{P}(v_j | D) = \frac{|\{< \ldots, v_j >\}|}{|D|}$$

$$\hat{P}(a_i | v_j, D) = \frac{|\{< \ldots, a_i, \ldots, v_j >\}|}{|\{< \ldots, v_j >\}|}$$

Proportion of times you see that class above the total.

Typical solution is Bayesian estimate with prior estimates (p-prior, m-weight)

$$\hat{P}(a_i | v_j, D) = \frac{|\{< \ldots, a_i, \ldots, v_j >\}| + mp}{|\{< \ldots, v_j >\}| + m}$$

$$
\begin{aligned}
P(PlayTennis = yes) = P(y) = 9/14 = 0.64 \\
P(PlayTennis = no) = P(n) = 5/14 = 0.36
\end{aligned}
$$

$$
\begin{aligned}
P(Wind = strong|y) &= 3/9 = 0.33 \\
P(Wind = strong|n) &= 3/5 = 0.60
\end{aligned}
$$

$$\ldots$$

$$
\begin{aligned}
P(y)\ P(sun|y)\ P(cool|y)\ P(high|y)\ P(strong|y) = .005 \\
P(n)\ P(sun|n)\ P(cool|n)\ P(high|n)\ P(strong|n) = .021
\end{aligned}
$$

$$\rightarrow v_{NB} = n$$

The strong wind influence more to say no

5.3 Learn to classify text

A set of documents ad input and a learn target function f: Docs → {c1,..,ck}

We compute a vocabulary V = {wk} (size n) with all the words appeared

Representations:

1. Boolean features: 1 if appear 0 otherwise (Multivariate Bernoulli)
2. Ordinal features: number of occurrences (Multinomial)

5.3.1 With Naive Bayes approach

Compute a Data set D = {<di, ci>} di documents

$$c_{NB} = \underset{c_j \in C}{\mathrm{argmax}}\, P(c_j|D) \prod_i P(d_i|c_j, D) \qquad P(d_i|c_j, D) = \prod_{i=1}^{length(d_i)} P(a_i = w_k|c_j, D)$$

where P(ai = wk |cj) is probability that word in position i is wk , given cj

Multi-variate Bernoulli Naive Bayes distribution

n-dimensional vector 1 if word wk appears in document d, 0 otherwise

$$P(d|c_j, D) = \prod_{i=1}^{n} P(w_i|c_j, D)^{I(w_i \in d)} \cdot (1 - P(w_i|c_j, D))^{1 - I(w_i \in d)}$$

$$\hat{P}(w_i|c_j, D) = \frac{t_{i,j} + 1}{t_j + 2}$$

$t_{i,j}$: number of documents in D of class c_j containing word w_i
t_j: number of documents in D of class c_j
1, 2: parameters for Laplace smoothing

Multinomial Naive Bayes distribution

n-dimensional vector with number of occurrences of word wi in document d

P(d|cj ,D) = Mu(d; n,) = . . .

$$\hat{P}(w_i|c_j, D) = \frac{\sum_{d \in D} tf_{i,j} + \alpha}{\sum_{d \in D} tf_j + \alpha \cdot |V|}$$

$tf_{i,j}$: term frequency (number of occurrences) of word w_i in document d of class c_j
tf_j: all term frequencies of document d of class c_j
α: smoothing parameter ($\alpha = 1$ for Laplace smoothing)

Algorithm (using Bernoulli)

Estimate $\hat{P}(c_j)$ and $\hat{P}(w_i|c_j)$ using *Bernoulli distribution.*

LEARN_NAIVE_BAYES_TEXT_BE(D, C)

- $V \leftarrow$ all distinct words in D
- for each target value $c_j \in C$ do
  - $docs_j \leftarrow$ subset of D for which the target value is c_j
  - $t_j \leftarrow |docs_j|$: total number of documents in c_j
  - $\hat{P}(c_j) \leftarrow \frac{t_j}{|D|}$
  - for each word w_i in V do
    - $t_{i,j} \leftarrow$ number of documents in c_j containing word w_i
    - $\hat{P}(w_i|c_j) \leftarrow \frac{t_{i,j}+1}{t_j+2}$