CA684 Jia Yuan Yu

3: Classifier selection

DCU March 7, 2014

These notes are based on the book "A Probabilistic Theory of Pattern Recognition."

The order of presentation of these sections in class may be different. First, we present more classification rules, then we consider the problem of selecting classifier functions from a deterministic class of functions.

1 Partitioning rules

Partition $\mathbb{R}^d$ into sets $\mathcal{A} = \{A_1, A_2, \ldots\}$ Partition rules classify according to majority voting in each set:

$$g_n(x) = \begin{cases} 1, & \text{if } \sum_{i=1}^n 1_{[Y_i=1]} 1_{X_i \in A(x)} > \sum_{i=1}^n 1_{[Y_i=0]} 1_{X_i \in A(x)}, \\ 0, & \text{otherwise,} \end{cases}$$

where $A(x)$ denotes the set containing x. A good partition has sets small enough to detect local changes in the label, but large enough to contain enough observation points.

For example, a cubic histogram rule partitions $\mathbb{R}^d$ into cubes of the same size. More generally, the partition can even depend on the observations $\{X_i\}$ and the number of training samples n. In particular, if g_n uses a partition of $\mathbb{R}^d$ into cubes of edges with length h_n, then the cubic histogram rule is strongly consistent if $h_n \to 0$ and $nh_n^d \to \infty$ as $n \to \infty$.

2 Kernel rules

A kernel function is a function $K : \mathbb{R}^d \to \mathbb{R}$ that is usually non-negative and with a single peak at the origin. Given a fixed parameter $h > 0$, a kernel classification rule is defined as:

$$g_n(x) = \begin{cases} 1, & \text{if } \sum_{i=1}^n 1_{[Y_i=1]} K(\frac{x-X_i}{h}) > \sum_{i=1}^n 1_{[Y_i=0]} K(\frac{x-X_i}{h}), \\ 0, & \text{otherwise,} \end{cases}$$

Kernel rules are similar to, but different from cubic histogram rules and nearest-neighbour rules.

Under some conditions (K being regular, $h \to 0$ and $nh^d \to \infty$ as $n \to \infty$), the kernel rule is strongly consistent.

3 Maximum likelihood rules

Let $p = \mathbb{P}(Y_j = 1)$. Let f_0 and f_1 denote the corresponding conditional densities of X_j conditioned on $Y_j = 0$ and $Y_j = 1$. The likelihood function of the data $(X_1, Y_1), \ldots, (X_n, Y_n)$ is

$$\ell_n(p, f_0, f_1) = \prod_{i=1}^{n} [p f_1(X_i)]^{Y_i} \, [(1-p) f_0(X_i)]^{1-Y_i},$$

and the log-likelihood function is $\log(\ell_n)$.

The maximum likelihood classification rule is

$$g_n(x) = \begin{cases} 1, & \text{if } p^* f_1^*(x) > (1-p^*) f_0^*(x), \\ 0, & \text{otherwise,} \end{cases}$$

where

$$(p^*, f_0^*, f_1^*) \in \arg\max_{\bar{p}, \bar{f}_0, \bar{f}_1} \log \ell_n(\bar{p}, \bar{f}_0, \bar{f}_1).$$

4 Neural networks

Neural nets are all the rage today, partly because of their simplicity, the large amount of training data available, and the availability of powerful computers.

A neural network with no hidden layer is the linear classifier

$$\phi(x) = \begin{cases} 1, & \text{if } \sum_{i=1}^{d} c_i x^i + c_0 > 1/2, \\ 0, & \text{otherwise,} \end{cases}$$

where $\{c_i\}$ are fixed and given or trained from data.

A σ-neural network with one hidden layer of k hidden neurons is

$$\phi(x) = \begin{cases} 1, & \text{if } \psi(x) > 1/2, \\ 0, & \text{otherwise,} \end{cases}$$

where

$$\psi(x) = \sum_{i=1}^{k} c_i \sigma(\psi_i(x)) + c_0,$$

$$\psi_i(x) = \sum_{j=1}^{d} a_{i,j} x^i + a_{i,0}, \quad \text{for all } i = 1, \ldots, k,$$

where $\{a_{i,j}, c_i\}$ are coefficients given or learned and σ is called a sigmoid function. Examples of sigmoid functions include the logistic sigmoid:

$$\sigma(z) = \frac{1 - e^{-z}}{1 + e^{-z}},$$

and the threshold sigmoid:

$$\sigma(z) = \begin{cases} 1, & \text{if } z > 0, \\ -1, & \text{otherwise.} \end{cases}$$

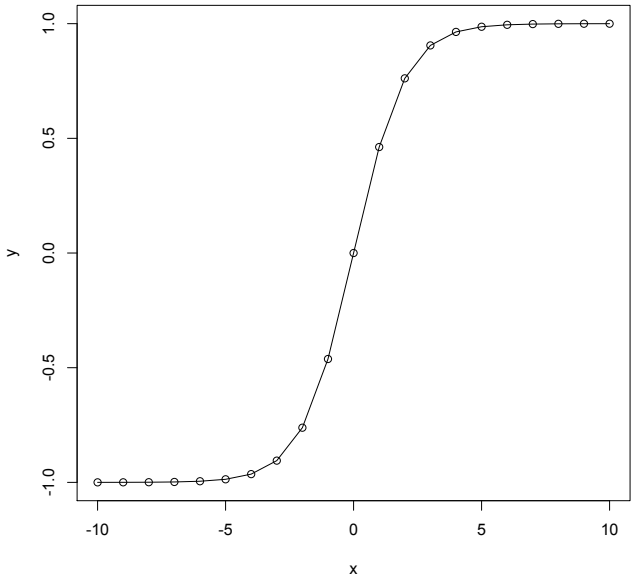

Figure 1: Logistic sigmoid.

4.1 Connection to partition rules

Consider the first (input) layer of the neural net. Let $\psi_1, \ldots, \psi_k$ denote the linear functions of the neurons of the first layer. Each ψ_j defines a hyperplane that partitions the observations space $\mathbb{R}^d$. Let $x\mathbb{R}^d$ denote the input. Suppose that we put a threshold sigmoid at the output of each neuron in the first layer. Then, the vector $z \in \{+1, -1\}^k$ of outputs of these sigmoids give you the index of the cell of the partition to which the input belongs. The following layers of the neural net assigns a decision to each cell, which replaces the majority vote of partition classifiers.

It is straightforward to define neural networks with many hidden layers and many hidden neurons per layer. To give performance guarantees to neural networks, we need VC theory. To understand VC theory, we first need some concentration inequalities.

5 Concentration inequalities

Concentration inequalities describe the phenomenon of sequences of random variables taking values near the mean, especially in the limit. For sums of i.i.d. Bernoulli random variables, we can consider the binomial distribution, which can be approximated by a normal distribution. The tail of the normal distribution is very "light" and can be evaluated accurately (cf. the Q- or erfc-functions), e.g., for $N(0, 1)$, we habe

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-u^2/2} \mathrm{d}u.$$

However, integrals are pesky, so it's nice to use closed-form bounds instead.

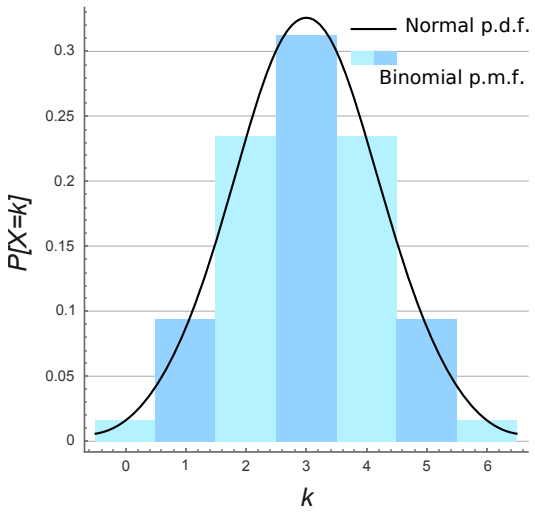

Figure 2: Source: Wikipedia.

Concentration inequalities give upper-bounds on the weight in the tail of sums of i.i.d. and not-necessarily Bernoulli random variables. One of the most useful concentration inequalities is Hoeffding's Inequality.

Theorem 5.1 (Hoeffding)**.** *Let $X_1, X_2, \ldots$ be independent random variables, where each X_i takes values in the interval $[a_i, b_i]$, and have mean μ. Let*

$$\hat{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i.$$

Then, for every n and $\varepsilon > 0$:

$$\mathbb{P}\left(\left|\hat{X}_n - \mathbb{E}\hat{X}_n\right| \geq \varepsilon\right) \leq 2\exp\left(-\frac{2n^2\varepsilon^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

Proof. The proof uses a number of steps. First step: we show that if $\mathbb{E}X = 0$, $a \leq X \leq b$, and $s > 0$, then

$$\mathbb{E}e^{sX} \leq \exp(s^2(b-a)^2/8). \tag{1}$$

By convexity, for $x \in [a, b]$:

$$e^{sx} \leq \frac{b-x}{b-a}e^{sa} + \frac{x-a}{b-a}e^{sb}$$

Since $\mathbb{E}X = 0$,

$$\begin{aligned} \mathbb{E}e^{sX} &\leq \frac{b}{b-a}e^{sa} - \frac{a}{b-a}e^{sb} \\ (\text{let } p = -a/(b-a)) \quad &= (1 - p + pe^{s(b-a)})e^{-ps(b-a)} \\ (\text{let } u = s(b-a), \phi(u) = -pu + \log(1-p+pe^u)) \quad &= e^{\phi(u)}. \end{aligned}$$

The function ϕ is twice differentiable. By Taylor's Theorem with Lagrange form remainder, there exists $\theta \in [0, u]$ such that

$$\phi(u) = \phi(0) + u\phi'(0) + \frac{u^2}{2}\phi''(\theta) \leq u^2/8,$$

where the last inequality uses:

$$\phi'(z) = -p + \frac{p}{p + (1-p)e^{-z}},$$
$$\phi''(z) = \frac{p(1-p)e^{-z}}{(p + (1-p)e^{-z})^2} \leq 1/4,$$

where the last inequality holds for all possible values of z and p. The claim follows.

Second step: by Markov's Inequality (for non-negative random variables), we have

$$\mathbb{P}(X \geq \varepsilon) = \mathbb{P}(e^{sX} \geq e^{s\varepsilon}) \leq \frac{\mathbb{E}e^{sX}}{e^{s\varepsilon}}$$

for all $s > 0$. Let $S_n = \sum_{i=1}^n X_i$, we have

$$\begin{aligned}
\mathbb{P}(S_n - \mathbb{E}S_n \geq \varepsilon) &\leq \mathbb{E}\exp\left(s\sum_i (X_i - \mathbb{E}X_i)\right)/e^{s\varepsilon} \\
\text{(by indep.)} \quad &= e^{-s\varepsilon}\prod_i \mathbb{E}e^{s(X_i - \mathbb{E}X_i)} \\
\text{(by (1))} \quad &\leq e^{-s\varepsilon}\prod_i \mathbb{E}e^{s^2(b_i-a_i)^2/8} \\
&= e^{-2\varepsilon^2/\sum_i (b_i-a_i)^2},
\end{aligned}$$

by choosing $s = 4\varepsilon/\sum_i (b_i - a_i)^2$. Next, we apply the same argument to $\mathbb{P}(S_n - \mathbb{E}S_n \leq -\varepsilon)$ to obtain the same bound.

Finally, we use the union bound:

$$\begin{aligned}
\mathbb{P}(|S_n - \mathbb{E}S_n| \geq \varepsilon) &\leq \mathbb{P}(S_n - \mathbb{E}S_n \geq \varepsilon) + \mathbb{P}(S_n - \mathbb{E}S_n \leq -\varepsilon) \\
&= 2e^{-2\varepsilon^2/\sum_i (b_i-a_i)^2}.
\end{aligned}$$

By simple algebra, we obtain the claim. □

6 What are concentration inequalities good for?

Concentration inequalities are useful in giving guarantees when selecting classifiers from a deterministic class of classifiers.

Let $\mathcal{C} = \{\phi_k : \mathbb{R}^d \to \{0, 1\}, k = 1, \ldots, K\}$ denote a class of K classifiers that are given and fixed. When we write $\phi \in \mathcal{C}$, we mean one of the classifiers $\phi \in \{\phi_1, \ldots, \phi_K\}$. Let

$$\hat{L}_n(\phi) = \frac{1}{n}\sum_{i=1}^n 1_{[\phi(X_i) \neq Y_i]}$$

denote the empirical error frequency of $\phi \in \mathcal{C}$. This random variable $\hat{L}_n(\phi)$ is an estimator for the error probability

$$L(\phi) = \mathbb{P}(\phi(X_j) \neq Y_j).$$

Observe that

$$\mathbb{P}\left(\sup_{\phi\in\mathcal{C}} \left|\hat{L}_n(\phi) - L(\phi)\right| > \varepsilon\right) \leq \sum_{\phi\in\mathcal{C}} \mathbb{P}\left(\left|\hat{L}_n(\phi) - L(\phi)\right| > \varepsilon\right)$$

by the union bound. Observe that $\hat{L}_n(\phi)$ is a (normalized) sum of Bernoulli random variables, each with mean $L(\phi)$. Hence, we can apply Hoeffding's Inequality and obtain the following.

Theorem 6.1. *For every $\varepsilon > 0$, we have*

$$\mathbb{P}\left(\sup_{\phi\in\mathcal{C}} \left|\hat{L}_n(\phi) - L(\phi)\right| > \varepsilon\right) \leq 2Ke^{-2n\varepsilon^2}.$$

7 Preview of VC theory

We define:

$$\phi^* \in \arg\min_{\phi\in\mathcal{C}} \hat{L}_n(\phi) = \arg\min_{k=1,\ldots,K} \hat{L}_n(\phi_k).$$

Define also:

$$L_n(\phi^*) = \mathbb{P}(\phi^*(X_j) \neq Y_j \mid X_1, Y_1, \ldots, X_n, Y_n).$$

Let $\inf_{\phi\in\mathcal{C}} L(\phi)$ denote the lowest probability of error among classifiers in $\mathcal{C}$—think of this as a poor man's Bayes error L^*, which is equal to L^* if the Bayes classifier g^* happens to be in $\mathcal{C}$.

Lemma 7.1. *We have*

$$L_n(\phi^*) - \inf_{\phi\in\mathcal{C}} L(\phi) \leq 2\sup_{\phi\in\mathcal{C}} \left|\hat{L}_n(\phi) - L(\phi)\right|.$$

By the above lemma, the event

$$L_n(\phi^*) - \inf_{\phi\in\mathcal{C}} L(\phi) > 2\varepsilon$$

implies the event

$$\sup_{\phi\in\mathcal{C}} \left|\hat{L}_n(\phi) - L(\phi)\right| > \varepsilon.$$

Hence,

$$\mathbb{P}\left(L_n(\phi^*) - \inf_{\phi\in\mathcal{C}} L(\phi) > 2\varepsilon\right) \leq \mathbb{P}\left(\sup_{\phi\in\mathcal{C}} \left|\hat{L}_n(\phi) - L(\phi)\right| > \varepsilon\right).$$

Finally, by Theorem 6.1, we have

$$\mathbb{P}\left(L_n(\phi^*) - \inf_{\phi\in\mathcal{C}} L(\phi) > 2\varepsilon\right) \leq 2Ke^{-2n\varepsilon^2}.$$

This is a preview of VC theory.

8 Estimating error probability

We can extend the previous result from deterministic class $\mathcal{C}$ to a class $\mathcal{C}_n$ of classifiers learned from data $(X_1, Y_1), \ldots, (X_n, Y_n)$:

$$\mathcal{C}_n = \{g_n^1, \ldots, g_n^K\}.$$

However, we have to be careful to compute a different set of empirical error frequencies $\hat{L}_m(g_n^k)$ on a distinct sequence of samples, such as

$$(X_{n+1}, Y_{n+1}), \ldots, (X_{n+m}, Y_{n+m}).$$

9 Homework

Pick your favorite machine learning software package and a big data set. Pick two of your favorite classification rules, and study empirically the relation between the empirical error frequency $\hat{L}_n(g_n)$ and the sample size n. Do this by counting errors on the training data $\hat{L}_n(g_n)$, and on a separate set of n testing data points $\hat{L}_{n,n}(g_n)$. Repeat this over K simulation runs. You should end up with:

$$\hat{L}_n^1(g_n), \ldots, \hat{L}_n^K(g_n),$$
$$\hat{L}_{n,n}^1(g_n), \ldots, \hat{L}_{n,n}^1(g_n).$$

Repeat this again for other values of n, such as $2n, 3n, \ldots$

Make a plot of the average (over the K simulations) error frequency versus n, along with error bars for one-standard deviation (over these simulations).