How to choose a model?

Assume that there are k models $M_1 \dots M_k$ that could potentially explain data $(x_1, y_1) \dots (x_n, y_n)$.

What model should we choose?

Bayes formula suggest posterior probabilities

$$P_r[M_i \mid (x_1,y_1)\dots(x_n,y_n)] = \frac{P_r[(x_1,y_1)\dots(x_n,y_n) \mid M_i] \cdot P_r[M_i]}{P_r[(x_1,y_1)\dots(x_n,y_n)]}$$

One of these models will have the largest probability If I must choose only one, I should pick the one with highest probability.

Note that the choice might heavily depend on my prior knowledge encoded as $P_r[M_1] \dots P_r[M_k]$

Maximum likelihood principle

If I am impartial — have no preferences to models:

$$P_r[M_i] = \text{const}, \quad i = 1 \dots k$$

then I should always choose the model with highest likelihood

$$P_r[(x_1,y_1)\dots(x_n,y_n) \mid M_i] \xrightarrow[i]{} \max.$$

This choice principle is known as maximum likelihood principle. We formulated it in terms of finite number of models. However, we generalise it to infinite number of models:

$$P_r[(x_1,y_1)\dots(x_n,y_n) \mid M] \xrightarrow[M]{} \max$$

This can be viewed as a particular or a process where we always choose among finite set of models but the precision can be arbitrarily small.

Maximum likelihood estimate for coin flipping

Set of models

Let model M_α denote setting where $x_1 \dots x_n$ are independently drawn from Bernoulli distribution with bias α

$$P_r[x_1 \dots x_n \mid M_\alpha] = \alpha^k (1-\alpha)^{n-k}$$

where k is the number of ones in the series $x_1 \dots x_n$

Maximization task

$$P_r[x_1 \dots x_n \mid M_\alpha] \xrightarrow[\alpha]{} \max$$

$$\Updownarrow$$

$$F(\alpha) = \alpha^k (1-\alpha)^{n-k} \xrightarrow[\alpha \in [0,1]]{} \max$$

Solution $\frac{dF}{d\alpha} = 0 \iff \alpha^{k-1}(1-\alpha)^{n-k+1}\big(k(1-\alpha) - (n-k)\alpha\big) = 0$

$$\iff \alpha = \frac{k}{n}$$

Maximum likelihood estimate for lin. regression

Set of models

Let us consider the class of models $M_{a,b}$ in the form $x_1, \dots, x_n$ are drawn from the normal distribution $N(0,1)$ and $y_1 \dots y_n$ are computed as

$$y_i = a x_i + b + \varepsilon_i \quad \text{for } \varepsilon_i \sim N(0, \sigma)$$

Now note that under "independent draws" assumption:

$$P[(x_1,y_1)\dots(x_n,y_n) \mid a,b] = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x_i^2}{2}\right) \cdot \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left(-\frac{(y_i - a x_i - b)^2}{2\sigma^2}\right)$$

$$= \underbrace{\frac{1}{(2\pi)^n \sigma^n} \cdot \exp\left(-\sum_{i=1}^{n} \frac{x_i^2}{2}\right)}_{\text{constant wrt } a,b} \cdot \underbrace{\exp\left(-\sum_{i=1}^{n} \frac{(y_i - a x_i - b)^2}{2\sigma^2}\right)}_{F(a,b)}$$

Further remarks and analysis

Since the density $p(x_1,y_1)\dots(x_n,y_n)|a,b)$ contains terms that do not change when we vary parameters a and b we can solve the following maximisation task

$$F(a,b) = \exp\left(-\sum_{i=1}^{n}\frac{(y_i-ax_i-b)^2}{2\sigma^2}\right) \xrightarrow[a,b]{} \max$$

Since logarithm is monotonously growing:

$$F(a,b) \to \max \iff \log F(a,b) \to \max$$

Hence, we can maximise

$$-\sum_{i=1}^{n}\frac{(y_i-ax_i-b)^2}{2\sigma^2} \xrightarrow[a,b]{} \max$$

$$\Updownarrow$$

$$\sum_{i=1}^{n}(y_i-ax_i-b)^2 \xrightarrow[a,b]{} \min$$

Maximum likelihood and neural networks

Set of models

A fixed network structure with variable weights w. Let $f_w(x)$ denote the output of the network on input x

Again the distribution of x_i-s is not important.
As before, we define

$$y_i = f_w(x_i) + \varepsilon_i \quad \text{for } \varepsilon_i \sim N(0,\sigma)$$

As a result

$$\log p[(x_i,y_i)|w] = \text{const} - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-f_w(x_i))^2$$

and we must minimise

$$\sum_{i=1}^{n}(y_i-f_w(x_i))^2$$

Predictions and sanity checks

If our assumption about the model is correct then the actual value y that corresponds to x is distributed as

$$y \sim N(f_w(x),\sigma)$$

So we should output a probability distribution or a 95% confidence interval. For that we need to estimate σ. If we would know the model parameters then $y_i - f_w(x_i)$ would be sampled from $N(0,\sigma)$. Consequently

$$\sigma \approx \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-f_w(x_i))^2}$$

The latter allows us to draw 95% confidence intervals. You would expect 95% of training samples be inside of these. Secondly you could confirm that residuals are from normal distribution with qq-plot.

Going beyond white Gaussian noise

Normal distribution has light tails. Samples really rarely deviate from mean by three or more standard deviations.

If the data contains many outliers, then there are two principle possibilities
1) use error distribution with heavier tails
2) use explicit modelling of outliers.

Standard solution for the first option is to assume

$$\varepsilon_i \sim \text{Laplace}(0,\beta)$$

Then $\log p(y_i|x_i,\beta) = \text{const} - |y_i - f(x_i)|$ and then the maximum likelihood principle yields

$$\sum_{i=1}^{n}|y_i-f_w(x_i)| \xrightarrow[w]{} \min.$$

Maximum aposteriory principle

Sometimes we have extra information that makes some models more likely than the others

$$P_A[M_i] \neq \text{const}$$

Then the model with highest likelihood is suboptimal and we should choose the model with highest posterior probability

$$P_A[M_i \mid (x_1,y_1)\dots(x_n,y_n)] \xrightarrow[i]{} \max$$

This method is known as maximum aposteriori principle (MAP)

Linear regression and MAP

Recall that the restriction

$$\|w\|_1 = |w_0| + \dots + |w_k| \leq C$$

answer that $f(x) = w_1x_1 + \dots w_kx_k + w_0$ is bounded by C in the region $x_i \in [-1,1]$. Hence, if I know that the output values

$$|f(x_i)| \leq C \quad \text{for all } x_{ij} \in [-1,1]$$

then I should assign a prior

$$p(w) = \begin{cases} \text{const}, & \|w\|_1 \leq C, \\ 0, & \|w\|_1 > C. \end{cases}$$

Recall that the latter leaded to minimisation task

$$-\log p((x_1,y_1),\dots,(x_n,y_n)\mid w) + \alpha\|w\|_1 \xrightarrow[w]{} \min$$

Alternative interpretation to priors

If we take penalised cost function as log of posterior we can reverse the calculation and restore prior:

$$p[w \mid \text{Data}] = P[\text{Data} \mid w] \cdot \exp(-\alpha\|w_i\|) \cdot C$$

$$\Downarrow$$

$$p[w] = C_2 \cdot \exp(-\alpha\|w\|_1)$$

and thus we implicitly prefer solutions with low sum of absolute coefficients.

This linear regression algorithm is known as lasso and it leads to solutions with handful non-zero coefficients. There models are used when you need highly interpretable connection between inputs and outputs.

Ridge regression and MAP

Recall that the restriction

$$\|w\|_2^2 = w_0^2 + \dots + w_k^2 \leq C$$

answer that $f(x) = w_1x_1 + \dots + w_kx_k + w_0$ is bounded

$$|f(x)| \leq \epsilon \quad \text{for all } \|x\|_2^2 \leq 1.$$

Hence, if I want to bound the function in unit ball I should assign a prior

$$p(w) = \begin{cases} \text{const}, & \|w\|_2^2 \leq C \\ 0, & \|w\|_2^2 > C \end{cases}$$

The latter leads to the minimisation task

$$-\log p((x_1,y_1)\dots(x_n,y_n)\mid w) + \alpha\|w\|_2^2 \xrightarrow[w]{} \min$$

Linear classification via modelling

Assume that data points come from two distributions.

To be concrete assume that

$$x_i \sim \mathcal{N}(\mu_0, \Sigma_0) \quad \text{if } y_i = 0$$

$$x_i \sim \mathcal{N}(\mu_1, \Sigma_1) \quad \text{if } y_i = 1$$

As a result

$$p[(x_1,y_1)\dots(x_n,y_n) \mid \mu_0,\mu_1,\Sigma_0,\Sigma_1] = \prod_{i=1}^{n} \frac{1}{(2\pi)^{n/2}\det\Sigma_{y_i}} \exp\left(-\frac{1}{2}(x-\mu_{y_i})^T\Sigma_{y_i}^{-1}(x-\mu_{y_i})\right)$$

Again it makes sense to consider log-likelihood

$$\log p[(x_1,y_1)\dots \mid \mu_0,\mu_1,\Sigma_0,\Sigma_1] = \text{const} + \sum_{y_i=0} -\log\det\Sigma_0 - \frac{1}{2}(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0) + \dots$$

Further analysis

It is easy to see that the log-likelihood decomposes into two functions

$$\log p[(x_1,y_1)\dots \mid \Sigma_1] = F_0(\Sigma_0,\mu_0) + F_1(\Sigma_1,\mu_1) + \text{const}$$

where

$$F_0(\Sigma_0,\mu_0) = -n_0\log\Sigma_0 - \frac{1}{2}\cdot\sum_{y_i=0}(x_i-\mu_0)\Sigma_0^{-1}(x_i-\mu_0)$$

$$F_1(\Sigma_1,\mu_1) = -n_0\log\Sigma_1 - \frac{1}{2}\sum_{y_i=1}(x_i-\mu_0)\Sigma_1^{-1}(x_i-\mu_1)$$

where n_0 and n_1 are the counts of labels 0 and 1.

To maximise the log-likelihood we can separately maximise F_0 and F_1. The latter is equivalent of fitting normal distributions to multivariate data

Unsupervised clustering. The model

When we know the class labels y_i then it is easy to estimate $\mu_0,\mu_1,\Sigma_0,\Sigma_1$ using maximum likelihood principle and later use Bayes formula to assign probabilities

$$P_r[y=0 \mid x] \quad \text{and} \quad P_r[y=1 \mid x]$$

However, sometimes class labels are missing.

Then we have to update the model

$$P_r[y_i=0] = \delta_0 \qquad P_r[y_i=1] = \delta_1$$

and estimate the posterior distribution

$$P_r[\mu_0,\mu_1,\Sigma_0,\Sigma_1 \mid \text{data}]$$

Analysis in one-dimensional case

Note that

$$p[x_i \mid \text{model}] = \delta_0\cdot p[x_i \mid \mu_0,\sigma_0] + \delta_1 p[x_i \mid \mu_1,\sigma_1] \quad i = 1\dots n$$

and thus

$$\log p[x_1\dots x_n \mid \text{model}] = \sum_{i=1}^{n} \log\left(\delta_0\cdot p[x_i \mid \mu_0,\sigma_0] + \delta_1 p[x_i \mid \mu_1,\sigma_1]\right)$$

In principle, we could solve this by computing derivatives

$$\frac{\partial}{\partial\mu_0}P,\ \frac{\partial}{\partial\mu_1}P,\ \frac{\partial}{\partial\sigma_0}P,\ \frac{\partial}{\partial\sigma_1}P$$

and then applying some sort of gradient descent algorithm. However, this is analytically difficult.

We see more advanced gradient search algorithms in the following lectures