MTAT.03.227 Machine Learning

Maximum Likelihood and
Maximum A Posteriori Estimates

Sven Laur
University of Tartu
How to choose a model?

Assume that there are $k$ models $\mathcal{M}_1, \ldots, \mathcal{M}_k$ that could potentially explain the data $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$. What model should we choose?

- Bayes formula leads to posterior probabilities

$$
\Pr[\mathcal{M}_i|\mathcal{D}] = \frac{\Pr[(x_1, y_1), \ldots, (x_n, y_n)|\mathcal{M}_i]\Pr[\mathcal{M}_i]}{\Pr[(x_1, y_1), \ldots, (x_n, y_n)]}
$$

- If I have to choose only one model, then I should choose the one with the highest posterior probability.
- My choice depends on prior probabilities $\Pr[\mathcal{M}_1], \ldots, \Pr[\mathcal{M}_k]$. 
Consider a series of coin flips $\mathbf{x} = (1, 0, 1, 0, 1, 0, 1, 0)$ and potential bias parameter values $\alpha = \{0.0, 0.1, \ldots, 0.9, 1.0\}$. Then we can use the Bayes formula and find the $\alpha$ value with the highest probability.
Illustrative example

However, note that a different prior can lead to completely different results.
Maximum likelihood principle

If I have no background information to prefer one model to another then

\[
\Pr [\mathcal{M}_i] = const
\]

and thus

\[
\Pr [\mathcal{M}_i | \mathcal{D}] = const \cdot \Pr [(x_1, y_1), \ldots, (x_n, y_n) | \mathcal{M}_i]
\]

As a result I should choose a model that maximises \textit{likelihood}

\[
\Pr [(x_1, y_1), \ldots, (x_n, y_n) | \mathcal{M}_i]
\]

The same principle is also applicable if the number of models is infinite.
For the observation vector $\mathbf{x} = (1, 0, 1, 0, 1, 0, 1, 0)$, the assumption that $\alpha \in \{0.0, 0.1, \ldots, 0.9, 1.0\}$ provides a rather coarse choice of models ...
The assumption that $\alpha \in \{0.00, 0.01, \ldots, 0.99, 1.00\}$ is better ...
Illustrative example continued ...

The assumption that $\alpha \in \{0.000, 0.001, \ldots, 0.999, 1.000\}$ is even better ...
ML estimate for coin-flipping

**Set of models.** Let model $\mathcal{M}_\alpha$ denote settings where $x_1, \ldots, x_n$ are drawn from Bernoulli distribution (independent coin-tosses):

$$\Pr[x_1, \ldots, x_n|\mathcal{M}_\alpha] = \alpha^k (1 - \alpha)^{n-k}$$

where the $k$ is the number of ones in the series $x_1, \ldots, x_k$.

**Maximisation task.** To solve $F(\alpha) = \alpha^k (1 - \alpha)^{n-k} \to \max$ we have to solve

$$\frac{\partial F}{\partial \alpha} = \alpha^{k-1} (1 - \alpha)^{n-k-1} (k(1 - \alpha) - (n - k)\alpha) = 0$$

From which we get $\alpha = k/n$. 
ML and linear regression

Set of models. Let model $\mathcal{M}_{a,b}$ denote settings where $x_1, \ldots, x_n$ are drawn from the normal distribution $\mathcal{N}(0, 1)$ and $y_1, \ldots, y_n$ are computed as

$$y_i = ax_i + b + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2)$$

Now note that

$$p[\mathcal{D}|a, b] = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{x_i}{2} \right) \cdot \frac{1}{\sqrt{2\pi\sigma}} \cdot \exp \left( -\frac{(y_i - ax_i - b)^2}{2\sigma^2} \right)$$

$$\text{const} \cdot \exp \left( -\sum_{i=1}^{n} \frac{(y_i - ax_i - b)^2}{2\sigma^2} \right)_{F(a, b)}$$
For instance, models with parameters $y = x$ and $y = -x$ generate the following probability distribution when $\sigma = 0.5$. 
Further analysis

To maximise

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

we can solve the minimisation task

\[ \sum_{i=1}^{n} \frac{(y_i - ax_i - b)^2}{2\sigma^2} \rightarrow \min \]

with respect to \( a \) and \( b \). We have obtained justification for the standard formalisation of linear regression.
ML and neural networks

Set of models. The set of models is fixed with a the topology of neural network and weight vector \( w \). Let \( f_w(x) \) be the models response to the input \( x \). As before, we can define

\[
y_i = f_w(x_i) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2)
\]

Now note that

\[
p[\mathcal{D}|w] = \text{const} \cdot \prod_{i=1}^{n} \exp \left( -\frac{(y_i - f_w(x_i))^2}{2\sigma^2} \right)
\]

and thus we must solve the following optimisation task

\[
\sum_{i=1}^{n} (y_i - f_w(x_i))^2 \rightarrow \min.
\]
Illustrative example
Predictions and sanity checks

Assume that ML estimate is precise enough then residues

\[ \Delta_i = y_i - f_w(x_i) \]

must be roughly distributed according to \( N(0, \sigma) \) and we can estimate

\[
\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - f_w(x_i))^2}.
\]

Now we can compute 95% confidence intervals for predictions:

> Around 95% of predictions should be in the confidence interval
> Residues should have normal distribution.
Beyond Gaussian noise

Sometimes we have too many outliers in the data. Such points will have high leverage since the distribution has light tails. There are two alternatives

▷ Use heavy-tail error distribution instead of normal distribution
▷ Model outlier points separately.

Usually the problem is solved by using centred Laplace distribution

\[ p[\varepsilon|\beta] = \frac{1}{2\beta} \cdot \exp \left( -\frac{|\varepsilon|}{\beta} \right) \]

As a result, we get a minimisation task

\[ \sum_{i=1}^{n} |y_i - f_w(x_i)| \rightarrow \min \].
Quick hack to implement second approach

To implement the second strategy, we have to find outlier points. For that, we can cycle the following algorithm till convergence:

- Train a model on normal data points.
- Estimate standard deviation of the noise $\sigma$.
- Use $\sigma$ to compute 95% confidence intervals for each data point.
- Label all points outside the confidence interval as outliers.

More advanced techniques require mixture modelling and EM-algorithm.
Illustrative example

Model fits the data much better if we remove most obvious outliers.
Maximum a posteriori principle

Sometimes, we have extra background knowledge that makes some models more likely than the others:

\[ \Pr [M_i] \neq const \]

Then the model with largest likelihood is suboptimal choice and we should take a model with highest posterior probability

\[ \Pr [M_i | D] \to \max . \]

This method is known as maximum a posteriori principle.

In most cases, MAP estimates are defined so that they are numerically and statistically more stable than ML estimates.
MAP and linear regression

Let \( f(x) = w_1 x_1 + \ldots + w_k x_k + w_0 \). Then the restriction

\[
\|w\|_1 = |w_0| + \cdots + |w_k| \leq c
\]

guarantees that \( |f(x)| \leq c \) in the range \( x_i \in [-1, 1] \).

Hence, if I have background information that \( f \) is bounded in this range then I should assign prior

\[
p(w) = \begin{cases} 
\text{const} & \text{if } \|w\|_1 \leq c, \\
0 & \text{if } \|w\|_1 > c.
\end{cases}
\]
Further analysis

\[ p[w|D] = \begin{cases} 
const \cdot \exp \left( - \sum_{i=1}^{n} (y_i - f_w(x_i))^2 \right) & \text{if } \|w\| \leq c , \\
0 & \text{if } \|w\| > c , 
\end{cases} \]

Hence, we must solve the following minimisation trick

\[ \sum_{i=1}^{n} (y_i - f_w(x_i))^2 \rightarrow \min \quad \text{s.t.} \quad \|w\|_1 \leq c \]

Lagrange trick yields a non-constrained optimisation task

\[ \sum_{i=1}^{n} (y_i - f_w(x_i))^2 + \lambda \|w\|_1 . \]

This is known as \textit{lasso} regression method
MAP and ridge regression

Let \( f(x) = w_1 x_1 + \ldots + w_k x_k + w_0 \). Then the restriction

\[
\|w\|_2^2 = w_0^2 + \cdots + w_k^2 \leq c
\]

guarantees that \( |f(x)| \leq c \) in the range \( x_1^2 + \cdots + x_k^2 \leq 1 \). Hence, if I have background information that \( f \) is bounded in this range then

\[
p(w) = \begin{cases} 
  \text{const} & \text{if } \|w\|_2^2 \leq c \\
  0 & \text{if } \|w\|_2^2 > c 
\end{cases}
\]

as a result we get

\[
\sum_{i=1}^{n} (y_i - f_w(x_i))^2 + \lambda \|w\|_2^2.
\]
Going backwards

Penalised mean square errors can be traced back to a different prior

\[
p[M|D] = \text{const} \cdot \exp \left( - \sum_{i=1}^{n} (y_i - f_w(x_i))^2 - \lambda \|w\|^2 \right)
\]

\[
p[M|D] = \text{const} \cdot p[D|M] \cdot \exp \left( -\lambda \|w\|^2 \right)
\]

As the last term corresponds to centred multivariate normal distribution, we know that ridge regression assigns normal prior to \(w\).

- Prior is invariant under coordinate rotations
- Larger \(\lambda\) shrinks the distribution towards zero.