



# Machine Learning

---

## A First Learning Task



# Machine Learning

---

- Designing the elements of machine learning techniques is important for their operation and to understand what they actually learn
  - Representation
  - Evaluation / Performance estimation
  - Optimization



# Simple Learning Task

---

- Consider the following problem:
  - We pick up a pebble and paint a flat part of it in blue. Then we repeatedly toss it and mark the times when the blue side is at the bottom as success (S) and other tosses as fail (F).
    - Repeatedly tossing it, we get the following outcomes:  
S, F, S, S, F
  - Learn the characteristics of the pebble.



# Simple Learning Task

---

- What type of learning problem is this ?
  - Unsupervised model learning problem
    - Data does not contain desired output  
 $D = \langle S, F, S, S, F \rangle$
  - Supervised probabilistic prediction
    - Data has no input, only desired output  
 $\langle (, S), (, F), (, S), (, S), (, F) \rangle$
  - Interpretation depends what we consider the target output of the learning algorithm



# Simple Learning Task

---

- What is the answer to the model learning problem ?
  - Data:  $\langle S, F, S, S, F \rangle$
  - Most common answer:
    - Probability of success is 0.6
  - Why is that a good answer ?
    - What would the elements of a formal learning algorithm be that gives us this solution ?
      - Representation ?
      - Evaluation Function ?
      - Optimization technique ?



# Simple Learning Task

---

- Representation
  - Hypothesis space:
    - Set of Bernoulli Distributions  
Represents the probability of an outcome for independent tosses with two outcomes
  - Representation:
    - Parametric with probability of successful outcome as parameter,  $P(S)$



# Simple Learning Task

---

- Evaluation

- Performance function:

- Probability of generating the data

Bernoulli distribution gives answer for parameter  $P(S)$

$$P(D | P(S)) = P(S)^{n_s} (1 - P(S))^{(n - n_s)}$$

- Optimization criterion:

- Maximum Likelihood Estimation (MLE)

Find  $P(S)$  that maximizes  $P(<S, F, S, S, F> | P(S))$

$$P(S) = \arg \max_{\mu} \mu^3 (1 - \mu)^{(5-3)}$$



# Simple Learning Task

- Optimization

- Optimization approach:

- Analytic global optimization (since it is feasible)

- Convert to log likelihood:  $\hat{\mu} = \operatorname{argmax}_{\mu} P(D | \mu) = \operatorname{argmax}_{\mu} \ln(P(D | \mu))$

- Computer derivative:

$$\begin{aligned}\frac{d}{d\mu} \ln(P(D | \mu)) &= \frac{d}{d\mu} \ln\left(\mu^{n_S} (1 - \mu)^{(n - n_S)}\right) = \frac{d}{d\mu} (n_S \ln(\mu) + (n - n_S) \ln(1 - \mu)) \\ &= n_S \frac{1}{\mu} + (n - n_S) \frac{1}{1 - \mu} (-1) = n_S \frac{1 - \mu}{\mu(1 - \mu)} - (n - n_S) \frac{\mu}{\mu(1 - \mu)} \\ &= \frac{n_S - n\mu}{\mu(1 - \mu)}\end{aligned}$$

- Solve for extremum:

$$\frac{d}{d\mu} \ln(P(D | \mu)) = \frac{n_S - n\mu}{\mu(1 - \mu)} = 0 \quad \Rightarrow \quad P(S) = \mu = \frac{n_S}{n} = \frac{3}{5} = 0.6$$





# Simple Learning Task

---

- What is the answer to the model learning problem ?
  - Data:  $\langle S, F, S, S, F \rangle$
  - Most common answer:
    - Probability of success is 0.6
  - Why is that a good answer ?
    - Simple answer is the maximum likelihood estimate (MLE) for the underlying Bernoulli process
      - Maximizes probability that the model generates the data
  - Is this the best answer ?



# Simple Learning Task

---

- Is this the best answer ?
  - Depends on whether there is other information available
    - What happens if instead of picking up a pebble I picked up a random coin ?
      - Prior probability of coins would make  $P(S)=0.5?$  a better answer
  - A better answer might also tell me how likely this is the correct answer
    - Probability that the data actually represents the expected outcomes



# MLE and MAP

---

- MLE does not take into account the prior probability of the parameters

- Bayes law gets us to the posterior estimate

$$P(\mu | D) = \frac{P(D | \mu)P(\mu)}{P(D)}$$

- Maximum a posterior estimate (MAP) finds the best parameter considering the prior

$$\hat{\mu} = \operatorname{argmax}_{\mu} P(\mu | D) = \operatorname{argmax}_{\mu} \frac{P(D | \mu)P(\mu)}{P(D)} = \operatorname{argmax}_{\mu} (P(D | \mu)P(\mu))$$

- Learning with the posterior is Bayesian learning
  - MLE corresponds to MAP with uniform prior



# MAP

- To compute MAP a prior over the parameters has to be provided
  - Prior represents knowledge about the system
    - General priors make the problem very difficult to solve
    - Conjugate priors are special forms of priors that lead to a closed form for the posterior

- Conjugate prior for Bernoulli/Binomial is the Beta distribution

$$Beta(\mu | \beta_S, \beta_F) = \frac{\mu^{\beta_S-1} (1-\mu)^{\beta_F-1}}{B(\beta_S, \beta_F)} \quad , B(\beta_S, \beta_F) = \int_0^1 \eta^{\beta_S-1} (1-\eta)^{\beta_F-1} d\eta$$

- Behaves like the addition of samples to the dataset

$$\begin{aligned} \hat{\mu} &= \operatorname{argmax}_{\mu} (P(D | \mu) P(\mu)) = \operatorname{argmax}_{\mu} \left( \mu^{n_S} (1-\mu)^{(n-n_S)} \frac{\mu^{\beta_S-1} (1-\mu)^{\beta_F-1}}{B(\beta_S, \beta_F)} \right) \\ &= \operatorname{argmax}_{\mu} \left( \mu^{(n_S+\beta_S-1)} (1-\mu)^{(n-n_S+\beta_F-1)} \right) \end{aligned}$$



# Simple Learning Task

---

- How do I know how close my answer is ?
  - The data in the dataset can be biased
    - Sample and Sampling variance give us estimates for the reliability of the estimate calculated from the data
    - Hoeffding's inequality provides an upper bound on the variation and thus of the estimate for random sampling

- In the case of repeated Bernoulli trials with MLE:

$$P\left(\left|\hat{\mu} - \mu^*\right| \geq \varepsilon\right) \leq 2e^{-2n\varepsilon^2}$$

- Can use this to indicate reliability or to determine how many samples we need to get a good estimate

$$P\left(\left|\hat{\mu} - \mu^*\right| \geq \varepsilon\right) \leq 2e^{-2n\varepsilon^2} \leq \delta \quad \Rightarrow \quad n \geq \frac{\ln(2/\delta)}{2\varepsilon^2}$$



# Simple Learning Task

---

- If we can determine these bounds with probabilistic evaluation criteria we get PAC learning
  - Probably Approximately Correct (PAC)
  - PAC learning provides probabilistic guarantees on the quality of the results of learning
    - With some distributions and evaluation functions PAC learning results are known and achievable



# Simple Learning Task

---

- Could we have addressed this task differently ?
  - Treat the task as an expected value problem
    - Model outcomes as numbers,  $S=1, F=0$
    - $E[M]=P(S)*S=P(S)$
    - Find model that minimizes the error generating the data
  - Hypothesis space:
    - Value of the expected outcome
  - Representation:
    - Parametric with expected outcome as parameter
      - Gives us a probability in this special case since the expected value is the probability



# Simple Learning Task

---

- Evaluation function:
  - Squared error (or alternatively root square error (RSE) or root mean squared error (RMSE))

$$Error(\mu) = \sum_{x \in D} (x - \mu)^2$$

- Optimization criterion:
  - Minimum squared error

$$E[M] = \arg \min_{\mu} \sum_{x \in D} (x - \mu)^2$$





# Simple Learning Task

- Optimization:

- Analytic global optimization

$$\begin{aligned}\frac{d}{d\mu} \text{Error}(\mu) &= \frac{d}{d\mu} \sum_{x \in D} (x - \mu)^2 = \sum_{x \in D} \frac{d}{d\mu} (x - \mu)^2 = \sum_{x \in D} 2(x - \mu)(-1) \\ &= 2\left(|D|\mu - \sum_{x \in D} x\right) = 0\end{aligned}$$

$$\Rightarrow \hat{\mu} = \frac{\sum_{x \in D} x}{|D|} = \frac{3}{5} = 0.6$$

- In this case the answer is the same as for MLE
    - In general, squared error and MLE do not address the same objective and do not yield the same result
      - MLE optimizes the prediction in a binary fashion
      - Squared error optimizes prediction in terms of Cartesian similarity



# Simple Learning Task

---

- Did the previous solution give us the best estimate in terms of expected error ?
  - Task definition made some hidden assumptions
    - We are only interested in the error on the data, not the expected error on the models
      - These are only the same if the data sample does not provide biased estimates for the mean and variance
    - We did not take into account the prior likelihood of the models
  - If we want to get the most precise model we need to address expected model error
    - Address parameter error



# Simple Learning Task

---

- Evaluation function:
  - Expected squared error of the model parameter
    - “Similarity” of the learned to the real model

$$Error(\hat{\mu}) = \int_0^1 (\mu - \hat{\mu})^2 p(\mu | D) d\mu$$

- Optimization criterion:
  - Minimum squared error

$$\mu^* = \arg \min_{\hat{\mu}} \int_0^1 (\mu - \hat{\mu})^2 p(\mu | D) d\mu$$



# Simple Learning Task

- Optimization:

- Analytic global optimization

- Minimum of squared error is the expected value

$$\begin{aligned}\mu^* &= \arg \min_{\hat{\mu}} \int_0^1 (\mu - \hat{\mu})^2 p(\mu | D) d\mu = E[\mu] = \int_0^1 \mu p(\mu | D) d\mu \\ &= \int_0^1 \mu \frac{p(D | \mu) p(\mu)}{p(D)} d\mu = \int_0^1 \mu \frac{\mu^{n_s} (1 - \mu)^{n - n_s} p(\mu)}{\int_0^1 \mu^{n_s} (1 - \mu)^{n - n_s} p(\mu) d\mu} d\mu\end{aligned}$$

- If prior  $p(\mu)$  is uniform between 0 and 1 we get a Beta distribution

$$= \int_0^1 \mu \frac{\mu^{n_s} (1 - \mu)^{n - n_s}}{(n + 1)!} d\mu = \frac{n_s + 1}{n + 2} = \frac{4}{7}$$

- Note: The expected value is not the same as the MAP

- Expected value considers bias/variance in the mean of the dataset



# Takeaway

---

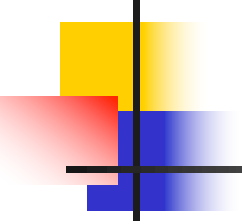
- Learning tasks can often be formulated in different ways by defining the data differently
  - Supervised vs. unsupervised learning sometimes just differs in the interpretation of the data
  - They can sometimes use the same evaluation/performance function
    - In supervised learning it has to be defined in terms of the target output while it is defined in terms of only the input data for unsupervised learning
  - They can use the same optimization approach



# Takeaway

---

- Different evaluation functions represent different characteristics and lead to different learning results
  - MLE and MAP consider “precision” in terms of a binary evaluation
  - Squared error considers “precision” in terms of a real valued similarity
  - It is important to choose the evaluation function carefully as it determines what is learned



# Designing/Choosing Machine Learning Algorithms

---

- To design/choose the right machine learning algorithm it is important to decide
  - What is the performance function
    - What should the algorithm learn ?
  - What type of learning problem is it
  - What is an efficient representation for the problem
    - What is the hypothesis space
    - How can the hypotheses be formulated
      - Parametric vs. non-parametric
      - What parameters
  - What is the most effective optimization approach