Introduction to Machine Learning
NPFL 054

http://ufal.mff.cuni.cz/course/npfl054

Barbora Hladká
hladka@ufal.mff.cuni.cz

Martin Holub
holub@ufal.mff.cuni.cz

Charles University,
Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics

Outline

- **Maximum likelihood estimation**
- **Course overview**

Probability vs. likelihood

The binomial distribution is the discrete probability distribution of the number of successes in a sequence of n independent yes/no experiments, each of which yields success with probability p, $X \sim Bin(n, p)$.

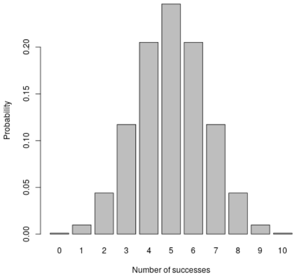
Task: Predict the outcome of each of 10 coin tosses

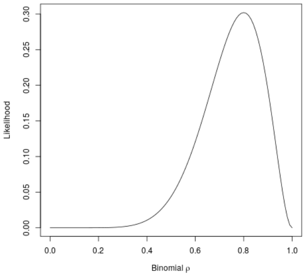
probability

$$\Pr(X = k | n = 10, p = 0.8)$$
$$\Pr(\mathrm{data}|\theta)$$

likelihood

$$\mathcal{L}(p|X = 8)$$
$$\mathcal{L}(\theta|\mathrm{data})$$



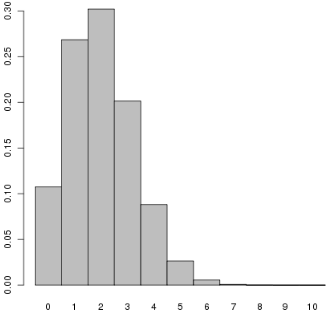


Binomial distribution

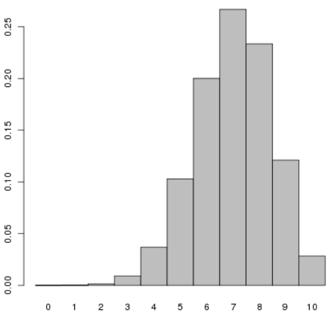
Probabilistic mass function

$$\Pr(X = k | n, p) = f(k; n, p) = \frac{n!}{k!(n-k)!} p^k (1-p)^{(n-k)}$$

$p = 0.2$



$p = 0.7$



Maximum likelihood estimation

- sample $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$

Assumption: $\mathbf{x}_1, \dots, \mathbf{x}_n$ are independent and identically distributed with an unknown probability density function $f(\mathbf{X}; \Theta)$

- Θ is a vector of parameters of the probability distribution $\Theta = \langle \theta_1, \dots, \theta_m \rangle$
- joint density function $f(\mathbf{x}_1, \dots, \mathbf{x}_n; \Theta) \stackrel{i.i.d.}{=} \prod_{i=1}^{n} f(\mathbf{x}_i; \Theta)$

We determine what value of Θ would make the data $\mathbf{X}$ most likely.

Maximum likelihood estimation

MLE is a method for estimating parameters from data.

Goal: identify the population that is most likely to have generated the sample.

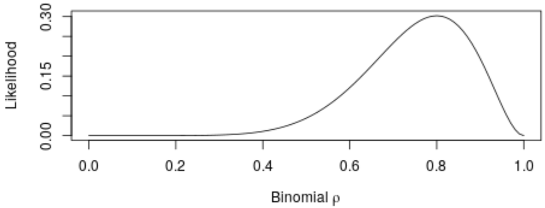
Likelihood function

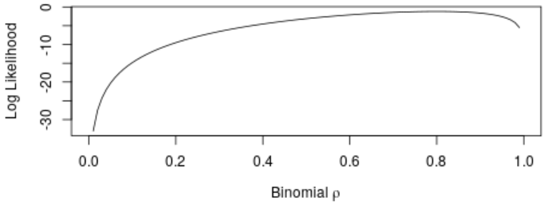
$$\mathcal{L}(\Theta|\mathbf{x}_1, \dots, \mathbf{x}_n) \stackrel{df}{=} \prod_{i=1}^{n} f(\mathbf{x}_i; \Theta) \tag{1}$$

Log-likelihood function

$$\log \mathcal{L}(\Theta|\mathbf{x}_1, \dots, \mathbf{x}_n) = \sum_{i=1}^{n} \log f(\mathbf{x}_i; \Theta) \tag{2}$$

Maximum likelihood estimation





Maximum likelihood estimation

Maximum likelihood estimate of Θ

$$\Theta^{\star}_{MLE} = \mathrm{argmax}_{\Theta} \log \mathcal{L}(\Theta|\mathbf{x}_1, \dots, \mathbf{x}_n) \tag{3}$$

Maximum likelihood estimation

MLE analytically

- Likelihood equation: $\frac{\partial \log \mathcal{L}(\Theta|X)}{\partial \theta_i} = 0$ at θ_i for all $i = 1, \ldots, m$
- Maximum, not minimum: $\frac{\partial^2 \mathcal{L}(\Theta|\mathbf{x})}{\partial \theta_i^2} < 0$

Numerically

- Use an optimization algorithm (for ex. Gradient Descent)

Maximum likelihood estimation
Binomial distribution

Estimate the probability p that a coin lands head using the result of n coin tosses, k of which resulted in heads. $\Theta = \langle p \rangle$

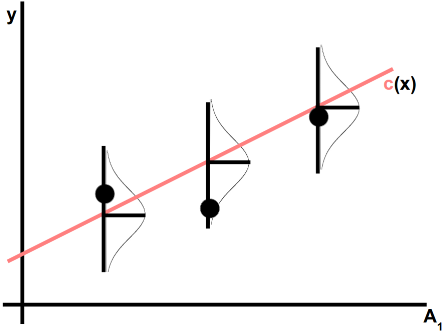
- $f(k; n, p) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}$
- $\mathcal{L}(p|n, k) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}$
- $\log \mathcal{L}(p|n, k) = \log \frac{n!}{k!(n-k)!} + k \log p + (n-k) log(1-p)$
- $\frac{\partial \log \mathcal{L}(p|n,k)}{\partial p} = \frac{k}{p} - \frac{n-k}{1-p} = 0$
- $\hat{p}_{MLE} = \frac{k}{n}$

Maximum likelihood estimation
Linear regression

Learn parameter estimates $\hat{\Theta}^\star$ from $Data = \{\langle \mathbf{x}_i, y_i \rangle, y_i \in \mathcal{R}, i = 1, ..., n\}$ using MLE.

Assumption: At each value of A_1, the output value y is subject to random error ϵ that is normally distributed $N(0, \sigma^2)$

$y_i = \Theta^\top \mathbf{x}_i + \epsilon_i$



Maximum likelihood estimation
Linear regression

- $\epsilon_i = y_i - \Theta^\top \mathbf{x}_i \sim N(0, \sigma^2)$
- probability density function of the Normal distribution

$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$\mathcal{L}(\mu, \sigma|\epsilon) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(\epsilon_i-\mu)^2}{2\sigma^2}}$$

$$\mathcal{L}(\Theta, \sigma|\mathbf{X}, \mathbf{y}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(y_i-\Theta^\top \mathbf{x}_i)^2}{2\sigma^2}}$$

Maximum likelihood estimation
Linear regression

$$\log \mathcal{L}(\Theta, \sigma | \mathbf{X}, \mathbf{y}) = \sum_{i=1}^{n} \log \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{(y_i - \Theta^\top \mathbf{x}_i)^2}{2\sigma^2}$$

$$\mathrm{argmax}_\Theta \log \mathcal{L}(\Theta, \sigma | \mathbf{X}, \mathbf{y}) = \mathrm{argmax}_\Theta \sum_{i=1}^{n} -\frac{1}{2\sigma^2}(y_i - \Theta^\top \mathbf{x}_i)^2$$

$$\mathrm{argmax}_{\boldsymbol{\Theta}} \log \mathcal{L}(\Theta, \sigma | \mathbf{X}, \mathbf{y}) = \mathrm{argmin}_\Theta \sum_{i=1}^{n} (y_i - \Theta^\top \mathbf{x}_i)^2$$

The minimum least square estimates are equivalent to the maximum likelihood estimates under the assumption that Y is generated by adding random noise to the true target values characterized by the Normal distribution $N(0, \sigma^2)$.

Maximum likelihood estimation
Logistic regression

Logistic regression models conditional probability using sigmoid function.

$$f(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{\Theta}^T\mathbf{x}}} = \Pr(y = 1|\mathbf{x})$$

Learn parameter estimates $\hat{\Theta}^\star$ from $Data = \{\langle \mathbf{x}_i, y_i \rangle, y_i \in \{0, 1\}, i = 1, ..., n\}$ using MLE.

Maximum likelihood estimation
Logistic regression

$$f(\mathbf{x};\Theta) = \Pr(y = 1|\mathbf{x})$$

$$\prod_{i=1}^{n} \Pr(y = y_i|\mathbf{x}_i) = \prod_{i=1}^{n} f(\mathbf{x}_i;\Theta)^{y_i}(1 - f(\mathbf{x}_i;\Theta))^{1-y_i}$$

$$\mathcal{L}(\Theta|\mathbf{X},\mathbf{y}) = \prod_{i=1}^{n} f(\mathbf{x}_i;\Theta)^{y_i}(1 - f(\mathbf{x}_i;\Theta))^{1-y_i}$$

$$\log \mathcal{L}(\Theta|\mathbf{X},\mathbf{y}) = \sum_{i=1}^{n} y_i \log f(\mathbf{x}_i;\Theta) + (1 - y_i)\log(1 - f(\mathbf{x}_i;\Theta))$$

$$\hat{\Theta}^{\star}_{MLE} = \mathrm{argmax}_{\Theta} \sum_{i=1}^{n} y_i \log f(\mathbf{x}_i;\Theta) + (1 - y_i)\log(1 - f(\mathbf{x}_i;\Theta))$$

Maximum likelihood estimation
Naïve Bayes classifier

$$\hat{y}^{\star} = argmax_{y_k \in Y} \Pr(y_k) \prod_{j=1}^{m} \Pr(x_j | y_k)$$

Maximum likelihood estimation
Naïve Bayes classifier

Categorical feature $\mathtt{A}_j$

Theorem

The Maximum likelihood estimates for NB take the form

- $\Pr(y) = \frac{c_y}{n}$ *where* $c_y = \sum_{i=1}^{n} \delta(y_i, y)$
- $\Pr(x|y) = \frac{c_{j_{x|y}}}{c_y}$ *where* $c_{j_{x|y}} = \sum_{i=1}^{n} \delta(y_i, y)\delta(\mathbf{x}_{i_j}, x)$

Maximum likelihood estimation
Naïve Bayes classifier

Continuous feature $\mathtt{A}_j$

Typical assumption, each continuous feature has a Gaussian distribution.

Theorem

The ML estimates for NB take the form

- $\overline{\mu_k} = \frac{\sum_{i=1}^{n} x_i^j \delta(y_i = y_k)}{\sum_{j=1}^{n} \delta(Y^j = y_k)}$
- $\overline{\sigma_k}^2 = \frac{\sum_{i=1}^{j} (x_i^j - \overline{\mu_k})^2 \delta(y_i = y_k)}{\sum_j \delta(Y^j = y_k)}$

$$\Pr(x|y_k) = \frac{1}{\sqrt{2\pi\overline{\sigma_k}^2}} e^{\frac{-(x - \overline{\mu_k})^2}{2\overline{\sigma_k}^2}}$$

Machine learning overview

machine learning = representation + evaluation + optimization

representation	**evaluation**	**optimization**
instances	**evaluation function**	**combinatorial**
k-NN	accuracy/error rate	greedy search
	precision, recall	
decision trees	ROC curve	
hyperplanes	**objective function**	**continuous**
Naïve Bayes	generative	*unconstrained*
	(conditional probability)	gradient descent,
Logistic regression	discriminative	maximum likelihood estimation
	(conditional probability)	*constrained*
SVM	margin	quadratic programming
Perceptron	mean square error	
graphical models		
Bayesian networks		
neural networks		

Machine learning overview

Task and data management

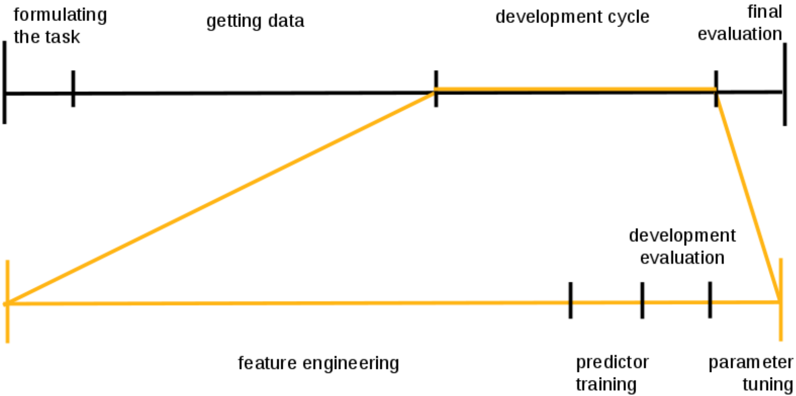
1. Time management
2. Formulating the task
3. Getting data
4. The more data, the better
5. Feature engineering
6. Curse of dimensionality

Methods and evaluation

7. Learning algorithms
8. Development cycle
9. Evaluation
10. Optimizing learning parameters
11. Overfitting
12. The more classifiers, the better
13. Theoretical aspects of ML

(1) Time management

How much time do particular steps take?



(2) Formulating the task

- Precise formulation of the task
- What are the objects of the task?
- What are the target values of the task?

(3) Getting data

- Gather data
- Assign true prediction
- Clean it
- Preprocess it
- Analyse it

(4) The more data, the better

If we don't have enough data

- **cross-validation** – The data set *Data* is partitioned into subsets of equal size. In the i-th step of the iteration, the i-th subset is used as a test set, while the remaining parts from the training set.
- **bootstrapping** – New data sets $Data_1$, ..., $Data_k$ are drawn from *Data* with replacement, each of the same size as *Data*. In the i-th iteration, $Data_i$ forms the training set, the remaining examples in *Data* form the test set

(5) Feature engineering

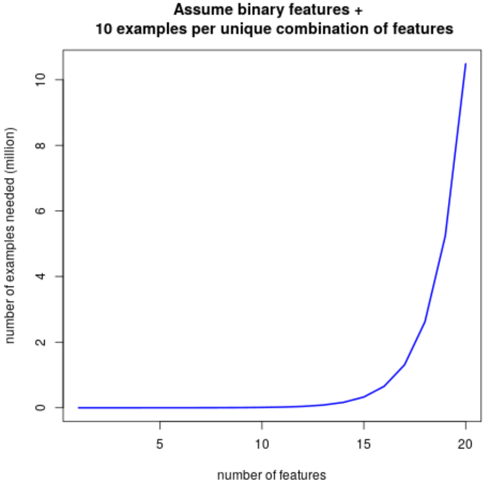
- Understand the properties of the objects
  - How they interact with the target value
  - How they interact each other
  - How they interact with a given ML algorithm
  - Domain specific
- Feature selection manually
- Feature selection automatically: generate large number of features and then filter some of them out

(6) Curse of dimensionality

- A lot of features $\longrightarrow$ high dimensional spaces
- The more features, the more difficult to extract useful information
- Dimensionality increases $\longrightarrow$ predictive power of predictor reduces
- The more features, the harder to train a predictor
- As the number of features (= dimensions) grows, the amount of data we need to generalize accurately grows exponentially
- **Remedy**: feature selection, dimensionality reduction

(6) Curse of dimensionality
Illustration

k binary features $\Rightarrow 10 \cdot 2^k$ examples needed



(7) Learning algorithms

Which one to choose?

First, identify appropriate learning paradigm

- Classification? Regression?
- Supervised? Unsupervised? Mix?
- If classification, are class proportions even or skewed?

In general, **no learning algorithm dominates all others on all problems**.

(8) Development cycle

- Test developer's expectation
- What does it work and what doesn't?

(9) Evaluation

Model assessment

- **Metrics** and **methods** for performance evaluation
  How to evaluate the performance of a predictor?
  How to obtain reliable estimates?

- **Predictor comparison**
  How to compare the relative performance among competing predictors?

- **Predictor selection**
  Which predictor should we prefer?

(10) Optimizing learning parameters

Searching for the best predictor, i.e.

- adapting ML algorithms to the particulars of a training set
- optimizing predictor performance

Optimization techniques

- Grid search
- Gradient descent
- Quadratic programming
- ...

(11) Overfitting

- bias
- variance

To avoid overfitting using

- cross-validation
- feature engineering
- parameter tuning
- regularization

(12) The more classifiers, the better

- **Build an ensemble of classifiers** using
  - different learning algorithm
  - different training data
  - different features
- **Analyze** their performance: complementarity implies potential improvement
- **Combine** classification results (e.g. majority voting).

Examples of ensemble techniques

- **bagging** works by taking a bootstrap sample from the training set
- **boosting** works by changing weights on the training set

(13) Theoretical aspects

Computational learning theory (CLT) aims to understand fundamental issues in the learning process. Mainly

- How computationally hard is the learning problem?
- How much data do we need to be confident that good performance on that data really means something? I.e., accuracy and generalization in more formal manner
- CLT provides a formal framework to formulate and address questions regarding the performance of different learning algorithms. Are there any general laws that govern machine learners? Using statistics, we compare learning algorithms empirically

References

- Pedro Domingos. A Few Useful Things to Know about Machine Learning. 2012.
  - https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf
- Pedro Domingos. Ten Myths About Machine Learning. 2016.
  - https://medium.com/@pedromdd/ten-myths-about-machine-learning-d888b48334a3

Summary of Examination Requirements

- Maximum likelihood estimations – likelihood function, loss function for logistic regression, MLE and least square method